



Universidad
Carlos III de Madrid

Departamento de Ingeniería Telemática

PROYECTO FIN DE CARRERA

DISEÑO E IMPLEMENTACION DE
UNA APLICACIÓN ANDROID
ORIENTADA A LA
ESQUEMATIZACIÓN DE OBRAS DE
TEATRO

Autor: Cristina Vidal Núñez

Tutor: Julio Villena Román

Leganés, 17 de marzo de 2014

Título: Diseño e implementación de una aplicación Android orientada a la esquematización de obras de teatro.

Autor: Cristina Vidal Núñez

Director: José Jesús García Rueda

EL TRIBUNAL

Presidente:

Secretario:

Vocal:

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 17 de marzo de 2014 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos:

A mi familia, por todo.

A José Jesús, por sus consejos y su paciencia.

A mi “familia madrileña”, Sergio, Alberto, Tania, Iván, Ezequiel, Irene, Elena, Mónica, Pili, Mafer, Fer, Vincent, Isa, Bruno, Elisa, Miguel, Taoufik, María, por estar ahí para mí, tanto en la Residencia como fuera de ella, por su comprensión y su apoyo.

A todos mis compañeros de teatro por sus ideas, su feedback y por hacer que estos años hayan sido tan divertidos.

Porque lo mejor del viaje es, sin duda, la gente que conoces por el camino.

Resumen:

El uso de las nuevas tecnologías en los campos artísticos abre un amplio abanico de posibilidades a la hora de facilitar el proceso creador.

El objetivo de este proyecto es crear una aplicación en Android para tablets y dispositivos móviles que permita a los ayudantes de dirección o creadores de espectáculos organizar sus montajes resumiendo escenas y guardando diagramas con las posiciones en el espacio de los diferentes actores.

Hemos escogido las tablets Android por emplear java como lenguaje al ser relativamente sencillo de usar y de distribución libre.

Palabras clave: Android, app, teatro, blocking.

Abstract:

The use of new technologies on artistic fields opens up a whole range of possibilities simplifying the creative process.

This Project goal is to create an Android app for tablets and mobile devices that allows the assistant director to organize their theatrical production by summarizing scenes and saving diagrams with the actor's locations on stage.

We choose Android tablets because the code of its apps is based on java, a relatively easy and free distribution language.

Key words: Android, app, Theater, blocking.

INDICE GENERAL

1.- Introducción: teatro y nuevas tecnologías	11
1.1 Introducción.....	11
1.2 Motivación:	11
1.3 Objetivos:	12
1.4 Estructura de la memoria:	12
2.- Estado del arte:	15
2.1 Introducción.....	15
2.2 Nuevas tecnologías como soporte al trabajo del director:.....	16
2.2.1 StageWrite	21
2.2.2 Theater Blocking.....	23
2.3 Android.....	24
2.3.1 Introducción	24
2.3.2 SQLite:.....	28
2.3.3 Programando en Android: MIT App Inventor	29
2.4 Conclusiones.....	30
3.- Diseño e implementación de la plataforma:	32
3.1 Requisitos	33
3.2 Diseño de la aplicación.....	34
3.3 Implementación	48
3.4 Pruebas.....	51
4.- Conclusiones y trabajos futuros:.....	55
4.1 Conclusiones.....	55
4.1 Trabajos futuros	55
5.- Presupuesto:	58
5.1 Costes de personal	58
5.2 Costes de material	58
5.3 Costes indirectos.....	59

5.4 Total	59
 6.- Anexos:	 61
6.1 Manual de uso.....	61
6.1.1 Instalación.....	61
6.1.2 Instrucciones de Teatro Virtual.....	61
6.2 Diagrama de clases	62
6.3 Glosario.....	67
6.4 Bibliografía.....	69

Capítulo 1

Introducción: teatro y nuevas tecnologías

1.1 Introducción

Las nuevas tecnologías se han convertido en una parte esencial del día a día y están presentes tanto en nuestro trabajo como en nuestro tiempo de ocio. De entre todos los campos, resulta especialmente interesante ver cómo éstas se fusionan y enriquecen numerosos aspectos de las ciencias sociales.

Tras varios años estudiando una ingeniería y varios más estudiando arte dramático comencé a preguntarme si ambos mundos están tan alejados como en un principio podría pensarse. ¿Es posible trabajar con las nuevas tecnologías apoyando el trabajo de actores y directores? Si bien es cierto que la industria teatral parece bastante tradicional y tiene fama de “analógica” cada vez son más los montajes que se atreven a integrar estos avances. Sería asimismo interesante investigar las nuevas tecnologías desde el punto de vista de la creación y organización de los espectáculos teatrales.

En resumen, este proyecto se plantea como un pequeño estudio sobre la relación entre el teatro y las nuevas tecnologías que nos dé una idea acerca de los procesos que podrían mejorarse usando estas últimas e intentar resolver los más básicos usando software.

1.2 Motivación:

Según la global mobile statistic de 2013 a día de hoy, existen 1.7 millones de aplicaciones para dispositivos inteligentes repartidas entre la iOS app store, Blackberry World, Google Play y Windows Phone. Este número no deja de crecer perteneciendo la mayor parte de ellas a Apple y Android, (unas 800.000 aplicaciones cada una).

Existen numerosas apps orientadas a ramas de las artes como la música, la literatura, la pintura o la fotografía. Es relativamente sencillo encontrar

aplicaciones que guardan grandes bases de libros, visitas virtuales a museos o herramientas de edición fotográfica; el teatro por su parte, no se encuentra tan integrado en este “mundo virtual”.

Las aplicaciones dirigidas al mundo del espectáculo son bastante escasas. Sin embargo existen muchos procesos propios del diseño de un montaje teatral que se podrían simplificar o incluso automatizar si se emplearan herramientas informáticas; es en esos aspectos en los que vamos a centrarnos.

1.3 Objetivos:

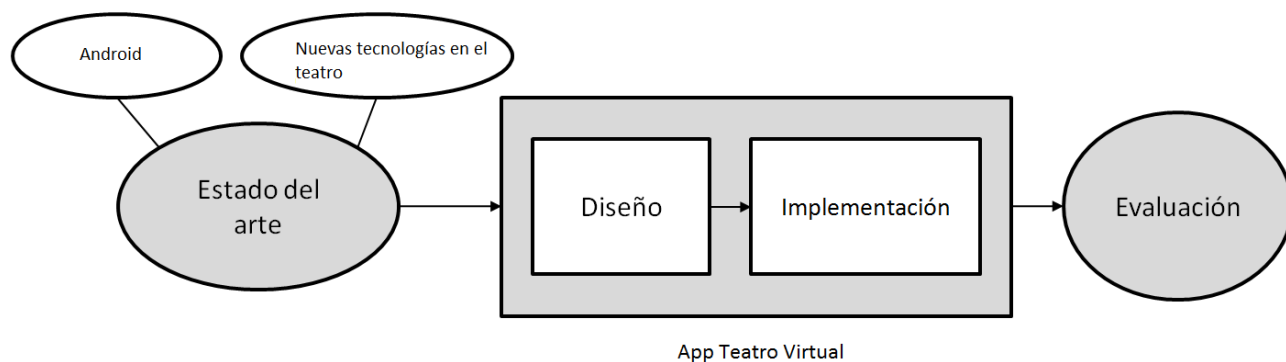
Tal y como se apunta en el anterior apartado el principal objetivo de este proyecto es diseñar y desarrollar una herramienta software que ayude a los profesionales del sector a simplificar parte de los procesos de la creación teatral, (en nuestro caso, una app para Android)

En base a este objetivo principal, se han definido los siguientes objetivos secundarios:

- Llevar a cabo una pequeña investigación sobre la relación actual entre el teatro y las nuevas tecnologías que ponga de manifiesto qué necesidades tienen los profesionales del sector (qué procesos tediosos y manuales podrían agilizarse con la ayuda de los nuevos dispositivos inteligentes) e investigar si estas están ya cubiertas por alguna otra aplicación.
- Realizar un estudio de las tecnologías existentes, un análisis de Android y del resto de lenguajes, (HTML, SQLite) que gravitan en torno a él a la hora de desarrollar una aplicación así como de sus entornos de desarrollo.

1.4 Estructura de la memoria:

Este documento presenta una estructura similar al proceso real que se siguió y que culminó con el app Teatro Virtual. En este primer tema hablamos de las necesidades del sector respecto a nuestra aplicación, aclaramos los objetivos del proyecto y explicamos el orden y contenido de los capítulos.



El capítulo dos se centra en el estado del arte dividiéndose en dos subapartados; el primero detalla el estudio realizado sobre la relación del teatro con las nuevas tecnologías, el segundo habla sobre Android y el uso que le vamos a dar a ese lenguaje.

En el capítulo tres profundizaremos en el diseño y desarrollo de la aplicación así como de su batería de pruebas. En el capítulo cuatro extraeremos conclusiones y analizaremos las posibles líneas de trabajo futuras y por último dedicaremos el capítulo cinco a hablar del presupuesto.

Capítulo 2

Estado del arte:

2.1 Introducción

Podemos dividir el desarrollo del estado del arte en dos fases diferenciadas. La primera consta un estudio acerca del estado actual del teatro respecto a las nuevas tecnologías, una pequeña investigación que nos ayuda a descubrir qué es verdaderamente necesario en una aplicación teatral permitiéndonos concretar el proyecto. La segunda es un estudio de los dispositivos de acceso existentes y sus tecnologías relacionadas para dar con aquel que se ajuste mejor a nuestras necesidades.

Antes de hablar de las nuevas tecnologías en el teatro se hace necesaria una definición clara de lo que entendemos por teatro (separándolo en la medida de lo posible del concepto performance, cine, tele, etc.)

Tal y como explica Peter Brook en su libro *El espacio vacío* (Brook, 1968) si atendemos a su definición clásica vemos que el acto teatral precisa de tres elementos básicos: actores, espectadores y un espacio físico donde se represente la obra. Este espacio tiene que ser común a emisores y receptores y ambos grupos deben encontrarse en él simultáneamente. Es fácil ver como esta descripción no hace justicia a muchas de las piezas contemporáneas; ¿Qué pasaría si actores y espectadores se encontrasen en salas distintas comunicándose mediante videoconferencia? ¿Y si usamos material pregrabado? ¿Puede existir un espectáculo donde los actores no sean seres vivos sino robots? ¿Y si actores y espectadores comparten el mismo espacio escénico pero los primeros están ocultos tras una estructura y solo se dejan ver a través de dispositivos electrónicos como cámaras de vídeo o fotográficas? Como podemos comprobar las preguntas pueden ser todo lo numerosas y complejas que queramos hacerlas, al igual que lo son las propuestas teatrales contemporáneas y los medios que en ellas se emplean.

El uso de nuevas tecnologías amplía las posibilidades que ofrece una representación a la vez que difumina su propia definición. Tal y como resume Anxo Abuín en su trabajo *Teatro y nuevas tecnologías*, Las artes escénicas, han asimilado en mayor o menor medida las múltiples herramientas tecnológicas, en especial las digitales, sin embargo las nuevas tecnologías aplicadas al teatro, todavía llaman demasiado la atención sobre sí mismas. Se impone, por tanto, una nueva forma de entender y de hacer teatro que sea más

flexible, más integradora y que se tome el tiempo de investigar las herramientas de las que dispone, no solo para representar una obra sino para planificarla y diseñarla.

Para este proyecto, hemos querido centrarnos en los aspectos propios de la preproducción teatral que puedan ser optimizados de forma sencilla y para los que todavía no se ha diseñado un software adecuado.

Si en un primer momento se pensó en una herramienta para reproducir obras de teatro de modo virtual con avatares 3D en movimiento, una reunión con un grupo de actores y creadores teatrales hizo que el proyecto se reorientara; lo realmente necesario es bastante más básico y a la vez más útil; en lugar de una simulación realista, se necesita una herramienta sencilla para organizar la obra y las posiciones de los personajes en las distintas escenas en lugar de tener que hacer los dibujos y diagramas manualmente. El programa tiene que ser más sencillo y útil que usar cuaderno y bolígrafo, intuitivo y tiene que proporcionar a los directores la suficiente libertad como para trabajar incluso los montajes menos convencionales. Es aquí donde entra en juego el concepto de blocking. Este término, aún sin traducción en castellano, fue acuñado en el siglo XIX por el director W. S. Gilbert y se refiere al proceso de planificación de las escenas atendiendo a la posición de los actores y elementos escenográficos. Si antes se hacía con réplicas en miniatura del escenario y empleando pequeños bloques de madera, ahora podemos modernizar el proceso, dejar lápiz y papel y desarrollar un software de acorde a estas necesidades.

Una vez descubierto qué necesitamos toca investigar cómo implementarlo. Es mucho más práctico que el director pueda moverse por el espacio físico real del teatro mientras crea sus esquemas, coloca a los actores o modifica los planos iniciales por lo que descartamos software para equipos de sobremesa en favor de aquellos que puedan instalarse en tablets y dispositivos móviles. Elegiremos el sistema operativo Android frente a iOS al ser de libre distribución y estar basado en el java, lenguaje en el que se ha basado gran parte de nuestra formación.

2.2 Nuevas tecnologías como soporte al trabajo del director:

Con el auge de las aplicaciones para dispositivos móviles, tanto para Android como para iOS los directores teatrales cuentan con varias apps, si bien es cierto que la mayoría son simplemente grandes bases de datos que almacenan obras escritas o grabaciones. También es necesario destacar que del escaso número de aplicaciones con las que nos encontramos, su práctica totalidad

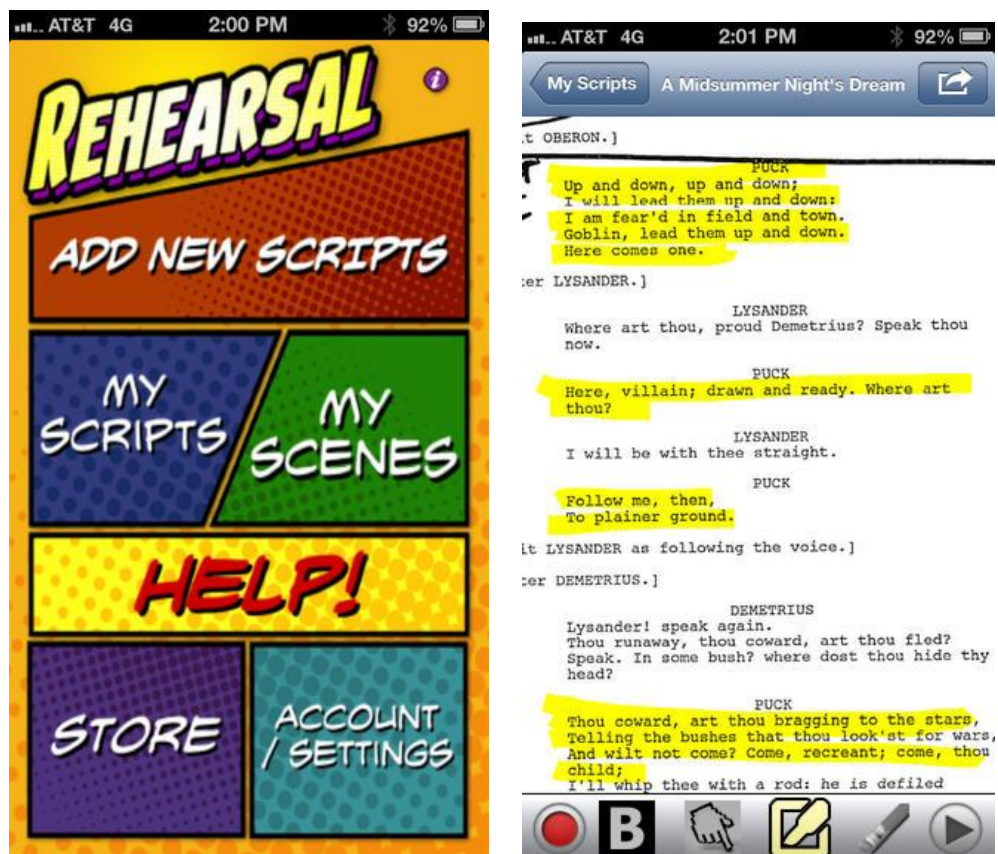
está en inglés, existiendo muy pocas herramientas orientadas a la producción teatral en nuestro idioma.

A continuación pasaremos a analizar las apps que tienen un perfil más técnico para hacernos una idea precisa de la situación del mercado, profundizando en aquellas que puedan parecerse más a la nuestra.

Aplicaciones para iOS:

Rehearsal:

Esta herramienta, pensada especialmente para los actores permite cargar un guion, resaltar las partes que más interesen y grabar audio por encima (muy útil para grabar las respuestas de los compañeros mientras se lee y reproducirlo para ensayar las propias). Su interfaz presenta un buen diseño y es de fácil manejo. Su precio es de 19.99 dólares.



Capturas de pantalla de Rehersal

Dramativity:

Esta aplicación te permite acceder a juegos teatrales, planificar clases y seminarios con material que se va actualizando semanalmente. También cuenta con una versión para Android. Descarga gratuita.

Sides:

Esta app también está pensada para que el actor ensaye sus líneas pero presenta más funcionalidades que Rehearsal. Además de grabar y resaltar el texto Sides permite anotar información de cada personaje (datos personales del actor, número de líneas etc.) y resaltar automáticamente sus intervenciones. Seleccionando el modo ensayo, la app tapa las líneas del actor y pausa las grabaciones para que ensaye sus frases a tiempo real. Descarga gratuita.



Capturas de pantalla de Sides

ScriptWrite:

Esta sencilla interfaz está pensada para facilitar la escritura de guiones. Cuenta con modo escritura y modo lectura, una base de datos de escenas y otra de personajes. Su precio es de 4.99 dólares.

StageWrite:

Una de las herramientas más útiles para todo aquel que quiera realizar un montaje teatral profesional y la que más se acerca a nuestra propuesta. La analizaremos con más detalle en el apartado 2.2.1

Scene Partner:

Esta aplicación está diseñada para ser usada por actores, escritores y los propios teatros.

Concebida como ayuda para memorizar diálogos dispone también con una versión para Android. Cuenta con un buen sistema de TTS (Text To Speech) con voces femeninas y masculinas así como con la opción de que los actores hagan sus propias grabaciones sobre los guiones. Aunque en un principio se orientaba únicamente al mundo anglosajón, su última versión, la 2.3, cuenta con voces francesas, italianas, alemanas, españolas y de chino mandarín.

La aplicación en sí es gratuita pero el precio de los guiones para cargar en la app ronda los 10 dólares y cada voz adicional cuesta 1.99 dólares. En esta aplicación los teatros disponen de un almacén virtual de obras que los actores pueden descargarse de forma segura y sin coste adicional.

Aplicaciones para Android:

Rehersal Assistant:

Bajo la apariencia de aplicación teatral nos encontramos con una simple grabadora de audio. Sus ventajas están en que permite almacenar las diferentes pistas por sesiones, ordenarlas como el usuario crea conveniente y mandarlas por mail. Aplicación gratuita.

Stage Manager:

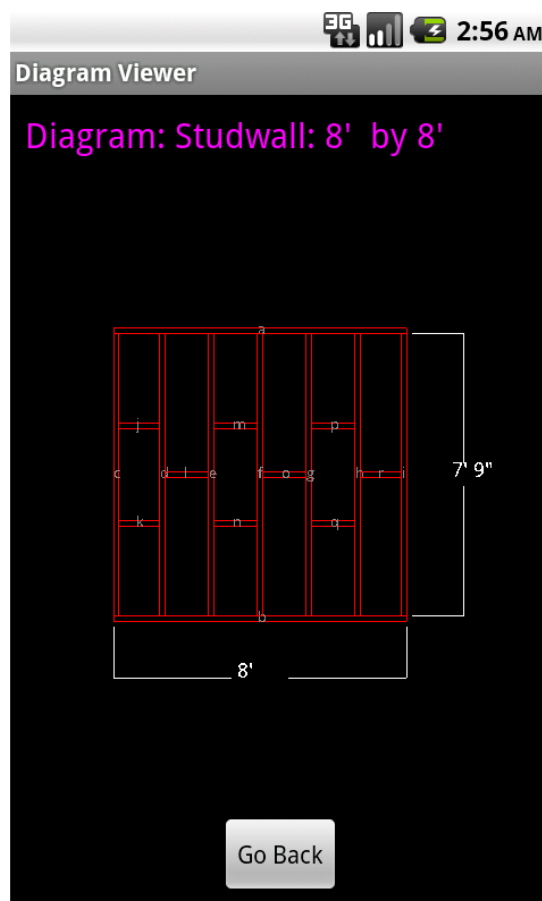
Básicamente ayuda a crear y guardar en orden los sonidos de un espectáculo lo que permitiría en el caso de shows sencillos prescindir de un técnico de sonido al ser el propio artista el que guarde previamente todos sus clips de audio.

Esta aplicación todavía tiene bastantes errores que depurar y no presenta ninguna mejora que compense su uso frente a otros programas convencionales de edición de audio. Aplicación Gratuita

Technical Theatre Assistant:

Esta aplicación es bastante más elaborada siendo de ayuda a la hora de crear la escenografía de una obra. Cuenta con varias opciones entre las cuales destaca una que te muestra paso a paso y con planos en 2D y 3D cómo crear distintos tipos de estructuras en función de las medidas y del material en que tengan que ser construidas.

También permite calcular las dimensiones de piezas de escenografía y de escaleras en función de las características que se introduzcan, calcular la fuerza de una cuerda dependiendo de sus medidas y su material y cuenta con una base de datos con información sobre los diferentes tipos de materiales usados para la construcción de escenografía. Su descarga es gratuita aunque conviene destacar que la compañía no se hace responsable del uso que se le dé a los cálculos realizados con su aplicación.



Ejemplo de cálculo con Theatre Assistant

StageHand:

Aplicación para realizar diversos tipos de cálculos útiles para los técnicos de iluminación o cualquier persona que necesite ayuda con los focos.

Entre otras cosas, calcula el diámetro de luz proyectado por los focos en función de la distancia a la que se encuentren y su ángulo, cuenta con un convertidor voltios/amperios y un calculador de color que muestra el resultado de las diferentes combinaciones de los 3 colores básicos. StageHand es gratuita pero su versión pro tiene un precio de 1.62 euros.

LD Tools:

Algo más básico que StageHand permite realizar ciertos cálculos con focos y circuitos eléctricos. Aplicación gratuita.

Swatchbook:

Es una base de datos con todas las gelatinas (filtros de color para focos) de las diferentes marcas que existen en el mercado y sus características. Su precio es de 0.75 euros.

Gel Books:

Similar a Swatchbook pero algo más básica. Requiere conexión a internet y es la primera aplicación que encontramos programada enteramente con App Inventor, herramienta que describiremos más adelante. Aplicación gratuita.

Theater Blocking:

Otra herramienta que también cuenta con premisas similares a la nuestra aunque está menos desarrollada. La comentamos con más detalle en el apartado 2.2.2

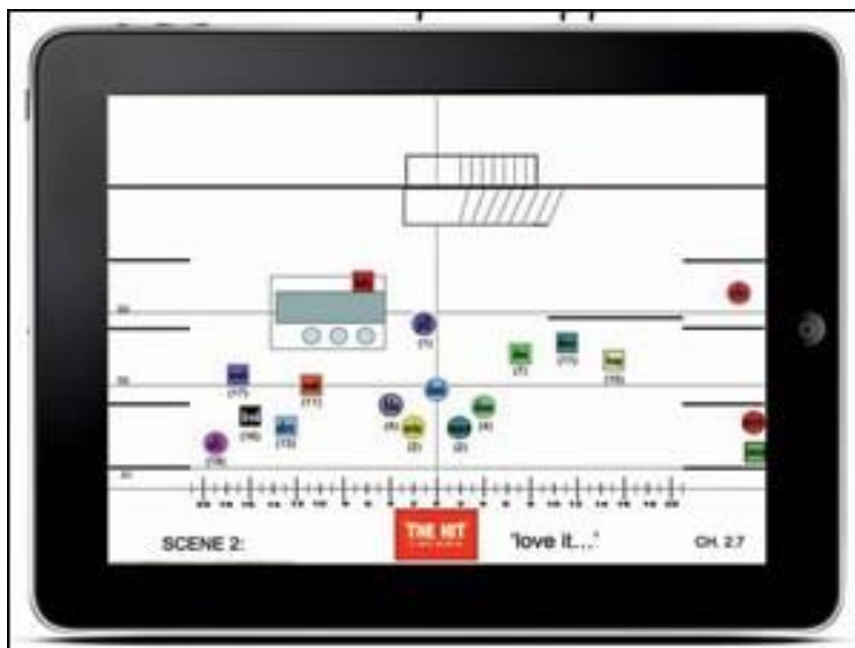
2.2.1 StageWrite

StageWrite fue concebida para crear gráficos espaciales dinámicos con los elementos materiales y humanos de una producción, lo que en inglés se conoce como blocking. Permite a coreógrafos, directores y ayudantes ahorrarse horas de trabajo, ya que solo tienen que crear el espacio e iconos de los actores una vez, y también les facilita el poder compartir sus esquemas con el resto de implicados como técnicos, orquesta o actores.

Esta aplicación permite elegir las dimensiones del escenario y dibujar aquellos elementos que vayan a estar fijos durante toda la función como cortinas o patas. También se pueden seleccionar piezas de escenografía o dibujarlas, cambiar sus características y señalar los caminos por los que se mueven los actores.

Una vez definido el espacio se pueden crear iconos de los personajes, con diferentes formas, colores y nombres e ir ubicándolos por el escenario cambiando su posición a medida que la escena avance y creando esquemas de movimiento dinámicos. Además de los gráficos la aplicación está diseñada para guardar notas y datos relativos al montaje.

Los gráficos creados se pueden reordenar, guardar en formato JPEG o PDF y enviar por correo. También se puede cambiar el punto de vista y pasar de un plano cenital a ver los diagramas desde el punto de vista del público o incluso de los actores.



Ejemplo de esquemas con StageWrite

Se trata de una aplicación muy útil si bien es cierto que puede resultar demasiado compleja debido al enorme número de funcionalidades que presenta. El precio, de casi 200 dólares, también es una desventaja ya que limita su uso a grandes producciones (grandes compañías como Sea Quest, el Circo del sol o musicales de Broadway). En resumen es una herramienta muy potente cuyo uso y coste solo compensa en caso de usarse para grandes montajes.

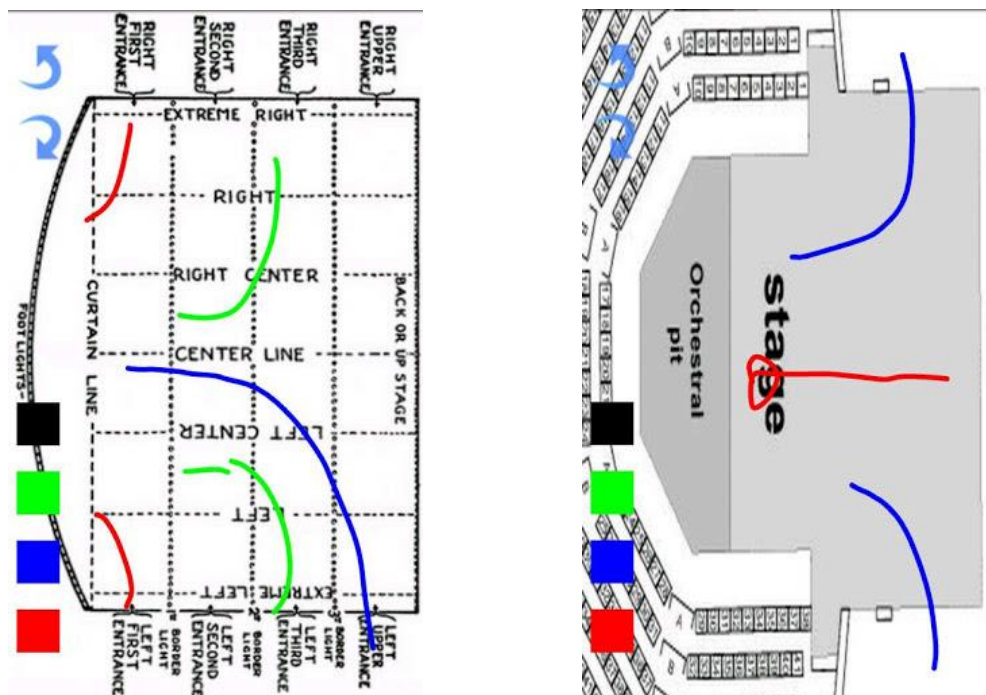
Si bien StageWrite tiene características muy interesantes, necesitamos que nuestra aplicación sea más sencilla, con un manejo intuitivo y al alcance de cualquier profesional del sector.

2.2.2 Theater Blocking

Esta aplicación sirve para seleccionar diferentes plantas de escenarios y dibujar encima de estas con distintos colores. Los dibujos se guardan en formato jpg en una carpeta por defecto pero la aplicación no permite cambiar su nombre ni ordenarlos de forma concreta lo que la convierte en un programa que podría acercarse a lo que queremos desarrollar pero que es muy poco práctico y presenta bastantes carencias.

A pesar de estar disponible en Google play esta no es una aplicación que cumpla con las características mínimas que la conviertan en un programa útil, apunta eso si ciertas ideas interesantes como la posibilidad de elegir el fondo en el que dibujas. Entre las opciones se encuentran tanto esquemas muy técnicos de las plantas (con el escenario dividido en cuadrantes) como otros que incluyen el foso de la orquesta e incluso los asientos del público. Esta última planta es muy interesante ya que son muchos los montajes teatrales que “rompen” la cuarta pared en beneficio de una interacción con el público.

También es una aplicación gratuita.



Ejemplo de esquemas con Theater Blocking

2.3 Android

2.3.1 Introducción

Android es un sistema operativo móvil basado en Linux, que junto con aplicaciones middleware está enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tablets, Google TV etc.

Tras estudiar los distintos tipos de sistemas nos decantamos por Android por varios motivos:

Tal y como comentábamos en el apartado 2.1 es importante que el sistema nos permita trabajar con dispositivos móviles, tarea para la cual Android es más que indicado.

El desarrollo de aplicaciones para Android requiere básicamente de conocimientos de java, que es un lenguaje con el que nos hemos familiarizado a lo largo de la carrera, y disponer del kit de desarrollo de software o «SDK» provisto por Google el cual se puede descargar gratuitamente. Para nuestra aplicación, también necesitaremos trabajar con bases de datos; usaremos SQLite, un lenguaje sencillo compatible con Android que explicaremos más adelante.

Android es de código abierto convirtiéndolo en un sistema operativo libre para que el desarrollador pueda modificarlo o mejorarlo a su gusto. Todas las aplicaciones se comprimen en formato APK y se pueden instalar fácilmente desde cualquier explorador de archivos en la mayoría de dispositivos. Esa es otra de las grandes ventajas de Android, al poder ser instalado en teléfonos de cualquier fabricante u otros dispositivos los usuarios cuentan con numerosos terminales de diferentes características y precios para elegir aquel que se adapte mejor a sus necesidades.

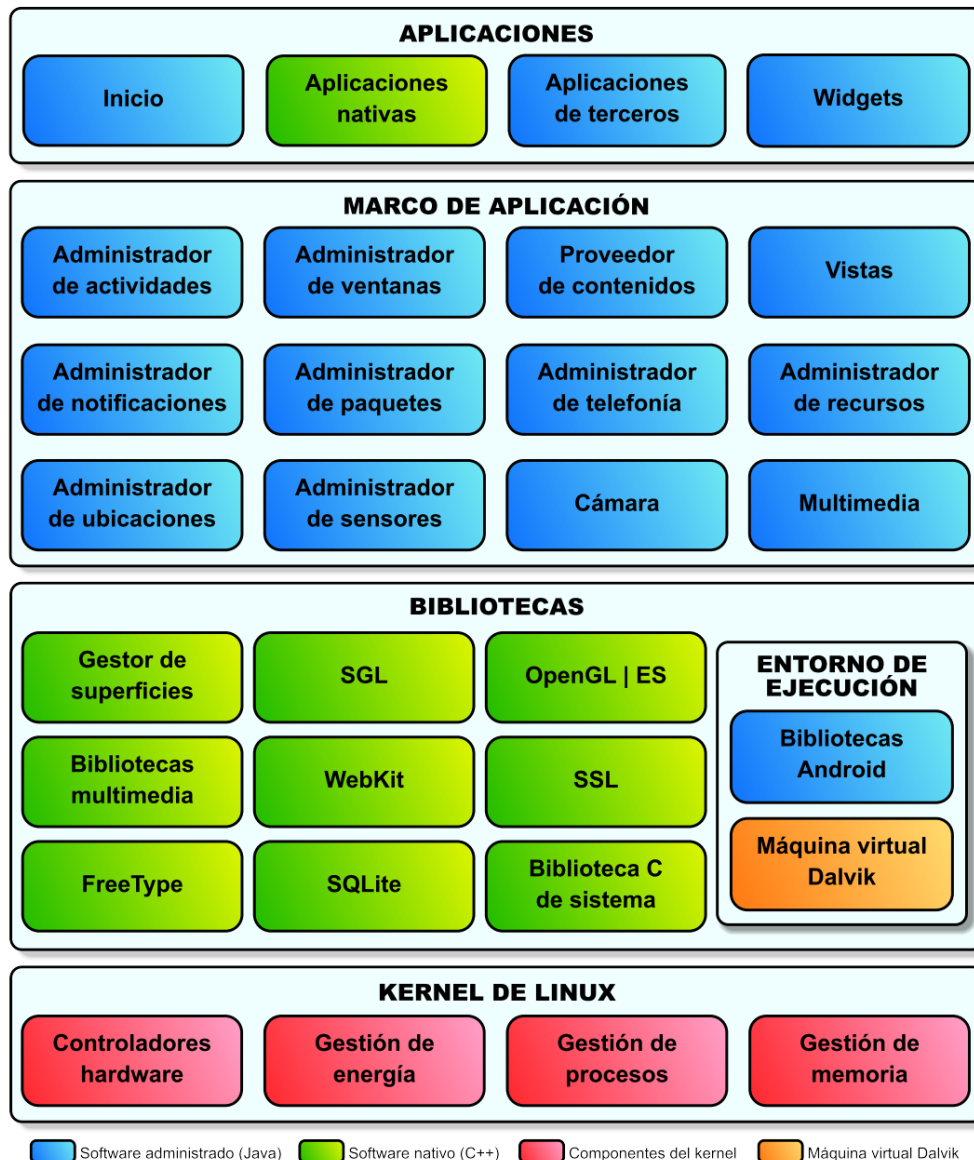
Los archivos apk pueden instalarse directamente en los dispositivos o descargarse gracias a aplicaciones como “play store” que permite acceder a la tienda virtual de software donde los desarrolladores publican sus aplicaciones Google Play.

A continuación vamos a detallar algunas de sus características más importantes:

- Cuenta con un framework de aplicaciones que permite reutilizar o sustituir las aplicaciones existentes.

- Dalvik VM es su máquina virtual, especialmente optimizada para dispositivos móviles. Android permite dar soporte a Java compilando su bytecode en un ejecutable Dalvik que posteriormente correrá en su máquina virtual.
- Soporte para el almacenamiento de datos estructurados mediante las librerías SQLite.
- Navegador integrado basado en el motor de código abierto WebKit.
- Capaz de procesar gráficos 2D y gráficos 3D.
- Multitarea. A diferencia de iOS, Android permite ejecutar varias aplicaciones simultáneamente, (las aplicaciones que se ejecutan en segundo plano también reciben ciclos de reloj).
- Multitáctil. Las pantallas Android son capaces de reconocer varios puntos de contacto simultáneamente.
- Google play. Este catálogo de aplicaciones permite su descarga e instalación en dispositivos Android sin necesidad de usar un PC.
- Soporte de múltiples formatos multimedia de audio, video e imagen.
- Telefonía GSM.
- Soporte para dispositivos hardware como cámaras, GPS, brújula o acelerómetros.
- Potente entorno de desarrollo que incluye un emulador, herramientas de depuración o un complemento para el IDE Eclipse.

La arquitectura del sistema operativo Android está formada por una pila de capas de software que interactúan entre sí siendo la más baja el Kernel o núcleo del sistema y siguiendo en este orden: el runtime de Android, las bibliotecas, el framework o marco de trabajo de las aplicaciones y las aplicaciones.



Esquema de la arquitectura Android.

El Kernel de Linux actúa como capa de abstracción entre el hardware y el resto de la pila de software. Android depende de Linux para servicios base del sistema como seguridad, gestión de memoria, gestión de procesos pila de red y modelo de controladores.

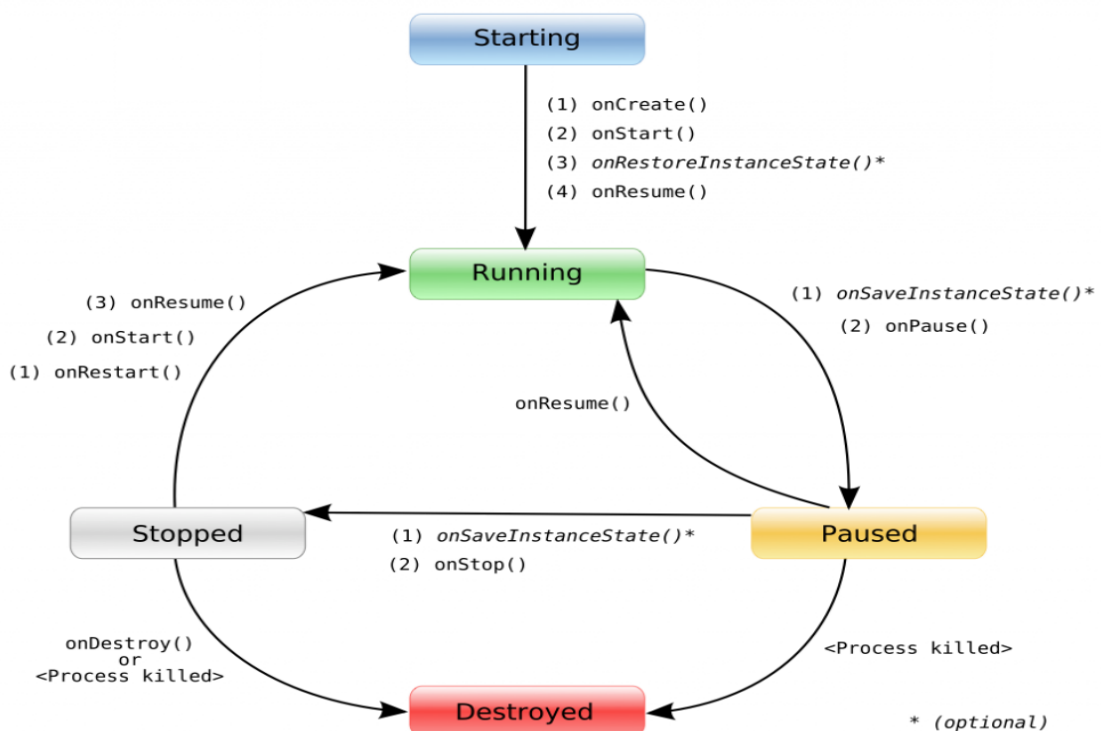
El Runtime de Android o entorno de ejecución incluye una serie de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik.

Dentro del conjunto de librerías en C y C++ con las que cuenta Android y que son usadas por varios componentes del sistema nos encontramos con System C library, bibliotecas de medios, de gráficos 3D y SQLite entre otras.

En lo que al marco de trabajo de las aplicaciones respecta los desarrolladores de apps tienen acceso a los mismos APIs del framework que usan las aplicaciones base. La arquitectura de Android permite reutilizar componentes o reemplazarlos. Las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas y navegador entre otros. Todas ellas están escritas en lenguaje de programación java.

Las aplicaciones Android son diferentes a las de los sistemas operativos tradicionales ya que estas apps funcionan bajo un esquema de actividades. Una actividad representa una pantalla, la cual interactúa con el usuario; las aplicaciones, por lo tanto, cuentan con una o más actividades.

Las actividades de la aplicación heredan de la clase Activity y presentan distintos estados, se pueden crear, destruir, suspenderse momentáneamente, etc.



Estados de la actividad con sus respectivos métodos

En conclusión, hemos escogido Android para llevar a cabo este proyecto por las siguientes razones:

- La naturaleza del lenguaje hace que se use en dispositivos móviles en lugar de ordenadores de sobremesa o portátiles, lo que beneficia a nuestra aplicación.
- Lenguaje de programación sencillo y herramientas asequibles.
- Política de desarrollo menos restrictiva.
- Terminales más baratos y de fácil acceso.

2.3.2 SQLite:

SQLite es un sistema de gestión de bases de datos relacional contenida en una biblioteca bastante manejable escrita en C.

A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente del programa principal sino que se convierte en parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones.

Pensamos que SQLite presenta la mejor combinación para trabajar con bases de datos en Android por los siguientes motivos:

- Android, posee una API por defecto para trabajar con SQLite. La clase SQLiteOpenHelper nos permite crear nuestra propia clase derivada de ella para comunicarnos y trabajar con este lenguaje de forma más o menos transparente.
- Herramienta muy estable al ser compatible con ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad)
- Es de dominio público por lo que se puede usar para cualquier propósito distribuyéndose libremente.
- Gran portabilidad; se ejecuta en numerosas plataformas y sus bases de datos pueden ser portadas sin ninguna configuración, (uso de un fichero por cada base de datos)
- El rendimiento de su base de datos es óptimo realizando operaciones de manera más eficiente y rápida que MySQL y PostgreSQL.
- Uso de registros de longitud variable: Generalmente los motores asignan una cantidad fija de espacio en disco para los campos de las tablas, SQLite disminuye el tamaño de bases de datos empleando sólo el espacio que se va necesitando lo que mejora su velocidad.

2.3.3 Programando en Android: MIT App Inventor

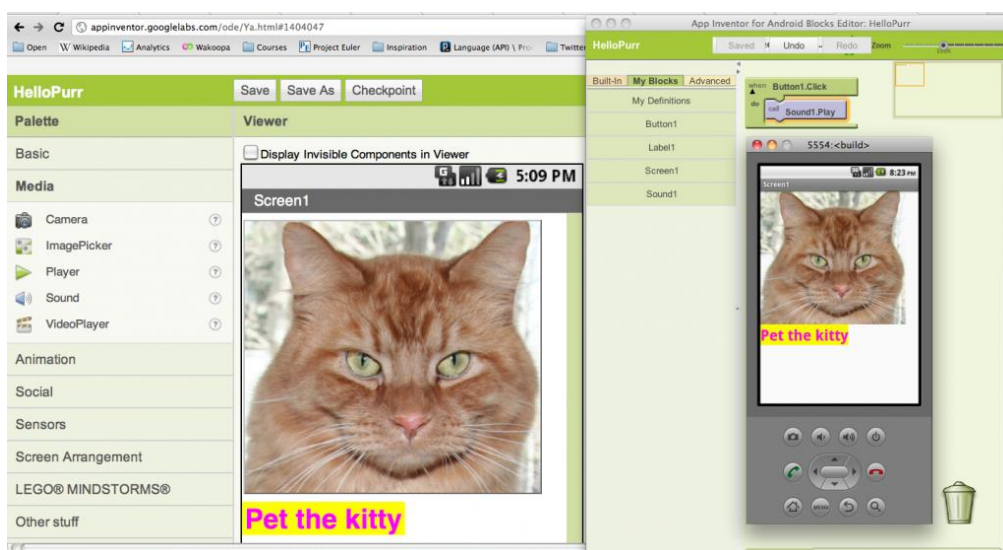
Si bien en esta aplicación se ha programado “manualmente” usando la herramienta de desarrollo Eclipse el uso de Android está tan extendido que conviene dedicar un tiempo a estudiar nuevos programas que surgen para ayudar a los docentes de la materia y a los programadores nóveles.

El App Inventor del Instituto Tecnológico de Massachusetts permite no solo crear una aplicación en sencillos pasos sino también explorar varios tutoriales y encontrar material de enseñanza.

Por un lado ofrece diferentes materiales y cursos ya elaborados para que los docentes puedan enseñar a alumnos de instituto o universitarios ofreciendo diferentes niveles, algunos de ellos con un lenguaje muy básico y sencillo para empezar a crear las primeras aplicaciones desde cero.

El App Inventor en sí, cuenta con una pantalla Designer y con un BlockEditor. El usuario diseña primero el aspecto que tendrá la aplicación; una vez creada la interfaz se puede definir el comportamiento de la aplicación de forma gráfica, uniendo diferentes “piezas” con instrucciones. Se puede tener conectada la aplicación a un dispositivo Android durante todo el proceso para ir viendo el avance de la misma.

App Inventor no está concebido para que sus usuarios trabajen con el código generado sino directamente con el resultado final. A pesar de ser una gran herramienta de aprendizaje y relativamente fácil de manejar para aplicaciones sencillas nuestra app es más compleja de lo que este programa te permite crear por lo que no se usará en este proyecto.



Interfaz de MIT App Inventor

2.4 Conclusiones

En conclusión y tras un meticuloso estudio del estado de las tecnologías en el ámbito teatral, y de las diferentes opciones para programar software, tomamos la decisión de desarrollar una herramienta que destaque por ser práctica y permitir de forma sencilla el diseño y creación de esquemas de movimiento para actores y escenografía dentro de un montaje teatral. Para ello emplearemos Android ayudados por el entorno de desarrollo de Eclipse y las bases de datos de SQLite.

El siguiente capítulo hablaremos en profundidad del diseño e implementación de nuestro Teatro Virtual, haciendo hincapié en la estructura interna de las aplicaciones Android para explicar posteriormente qué recursos hemos empleado y de qué manera.

Capítulo 3

Diseño e implementación de la plataforma:

Como explicamos en el capítulo 1 el objetivo principal de este PFC es crear un programa que permita guardar diferentes obras, con cada una de sus escenas y anotaciones y diseñar gráficos que muestren la posición de los actores en los distintos momentos de la escena. En este capítulo explicaremos las decisiones de diseño de Teatro Virtual y profundizaremos en aquellos aspectos de la implementación más significativos.

Son muchos los posibles desarrollos que podrían llevarse a cabo siguiendo estas pautas por lo que comenzaremos por analizar y explicar en detalle su funcionalidad para así poder sacar los requisitos de la aplicación, tanto a nivel de usuario como a nivel técnico acotando las posibilidades y diseñando el programa más adecuado.

Esta aplicación debe contar con una base de datos que permita al usuario almacenar obras de teatro. Dentro de cada una de estas obras deben poder crearse varias escenas con sus respectivos comentarios. Desde cada escena debería accederse a un editor de esquemas que permita situar a los actores en el espacio escénico.

Es fundamental que el diseño de la aplicación sea consistente con la dinámica real de las obras de teatro; dentro de cada escena los actores cuentan con libertad de movimientos por lo que es importante que la aplicación permita guardar varios esquemas por escena, que correspondería a las diferentes posiciones de los personajes. Con el fin de facilitar el trabajo del director convendría que durante la elaboración de los esquemas se mantuviera el último dibujo hecho pues muchas veces los cambios entre posiciones son sutiles y no interesa rehacer toda la escena desde cero. También es importante que esa pantalla cuente con todos los elementos necesario para poder reproducir una posición en condiciones; estos no deben de ser tan numerosos que dificulten la tarea del usuario y sí lo suficientemente genéricos como para que puedan amoldarse a cualquier requerimiento.

La presentación posterior tiene que ser dinámica y clara: convendría que tanto los esquemas como las notas que se tomen se muestren en la misma pantalla donde también se debería indicar la obra y escena a la que pertenecen. Esta última información debería de permanecer en la pantalla mientras se avanzan posiciones.

Teatro Virtual está pensado para ser usado tanto por actores como por directores aunque el uso que le den sea algo diferente. Los primeros son los encargados de completar toda la información que quieran transmitir mientras que los segundos probablemente solo necesiten visualizarlo por lo que necesitamos “atajos” que nos permitan llegar a la pantalla final sin necesidad de ir pasando por todos los editores de texto previos.

Con toda esta información sobre la mesa resulta sencillo enumerar los requisitos necesarios para Teatro Virtual.

3.1 Requisitos

Requisitos de usuario:

1. Capacidad de almacenamiento:

Es necesario que la aplicación almacene el título de las obras, las escenas, sus resúmenes, los dibujos de los esquemas y que además permita hacer cambios posteriores en cualquiera de esos campos.

2. Facilidad de uso:

Ha de ser una aplicación intuitiva y de uso sencillo, los usuarios no tienen por qué tener conocimientos informáticos avanzados.

3. Rapidez:

Tiene que ser una aplicación práctica que permita acceder a los resúmenes y esquemas de forma rápida sin necesidad de pasar por todas las pantallas.

4. flexibilidad:

Dada la naturaleza del trabajo que se va a realizar con esta aplicación, el programa tiene que conceder la suficiente libertad al usuario para que diseñe las obras a su antojo y no tenga, por ejemplo, que usar todas las opciones que ofrece el programa sino solo las que a él le interesen.

5. Interfaz adecuado:

La interfaz que se diseñe tiene que presentar las mismas características principales que el conjunto de la app: sencillez, rapidez y flexibilidad. No puede ser demasiado recargada y la función de cada botón y dibujo debe de quedar clara a simple vista.

Requisitos técnicos:

Una vez establecidos los requisitos de usuario vamos a centrarnos y a analizar los requisitos técnicos:

1. Tablet o dispositivo móvil:

Para ejecutar la aplicación necesitamos un dispositivo Android. Hemos optimizado la app para una tablet Bq Kepler de 8 pulgadas con Android 4.0.3, (Ice Cream Sandwich) aunque puede funcionar en prácticamente cualquier dispositivo. Nos decidimos por una tablet de estas características como punto de partida porque presenta un tamaño convencional, suficientemente grande como para que el usuario pueda desenvolverse por la app de forma cómoda pero con un precio asequible.

2. Android:

La versión de Android que hemos empleado para programar la aplicación ha sido la 2.2. Se ha elegido esta versión frente a otras más modernas porque es compatible con dispositivos que cuentan con versiones más nuevas y todavía se usa en muchos dispositivos no actualizados; de esta forma nos aseguramos de que nuestro programa llegue al mayor número de usuarios posibles.

3.2 Diseño de la aplicación

Antes de detallar las decisiones de diseño, es importante saber cómo se estructuran las obras de teatro. Aunque en este primer apartado explicaremos una división clásica y occidental, que nos es más cercana, en el diseño de la app seguirá primando la decisión de otorgar al usuario la mayor libertad posible para que cualquier tipo de montaje tenga cabida.

Estructura de una obra de teatro:

Las primeras unidades en las que se divide una pieza teatral “clásica” son los actos; cada uno de ellos engloba un conjunto de escenas que presentan características de acción similares (p.ej presentación, nudo y desenlace). Los actos no se tendrán en cuenta para estructurar nuestra aplicación pues carecen de utilidad práctica a la hora de crear diagramas de movimiento; nos

centraremos directamente en las escenas. El final o comienzo de una escena lo marca la salida o entrada de un actor al espacio, el cambio en el espacio o en el tiempo. Dado que actores y objetos se desplazan por el escenario en cada escena habrá que dividir esta unidad en otras más pequeñas: las posiciones o cuadros. Cada escena constará, pues, de varias posiciones que son las que se dibujan en los diagramas tradicionales y las que nosotros crearemos en nuestra aplicación.

Estructura de la aplicación:

En este apartado explicaremos en profundidad la estructura de nuestro proyecto; por un lado analizaremos su interfaz y por otro explicaremos la estructura interna de los proyectos Android.

Diseño de la interfaz del usuario

A continuación, explicaremos el diseño de Teatro Virtual realizando un recorrido ordenado por su interfaz y explicando las principales características de cada una de sus pantallas.

Antes de embarcarse en el diseño de la interfaz es fundamental tener claro no sólo las características con las que debe contar la aplicación sino también como se van a presentar; a qué usuarios va dirigida (los “actores” en este caso son los propios actores y los directores del montaje) y qué uso le dará cada uno.

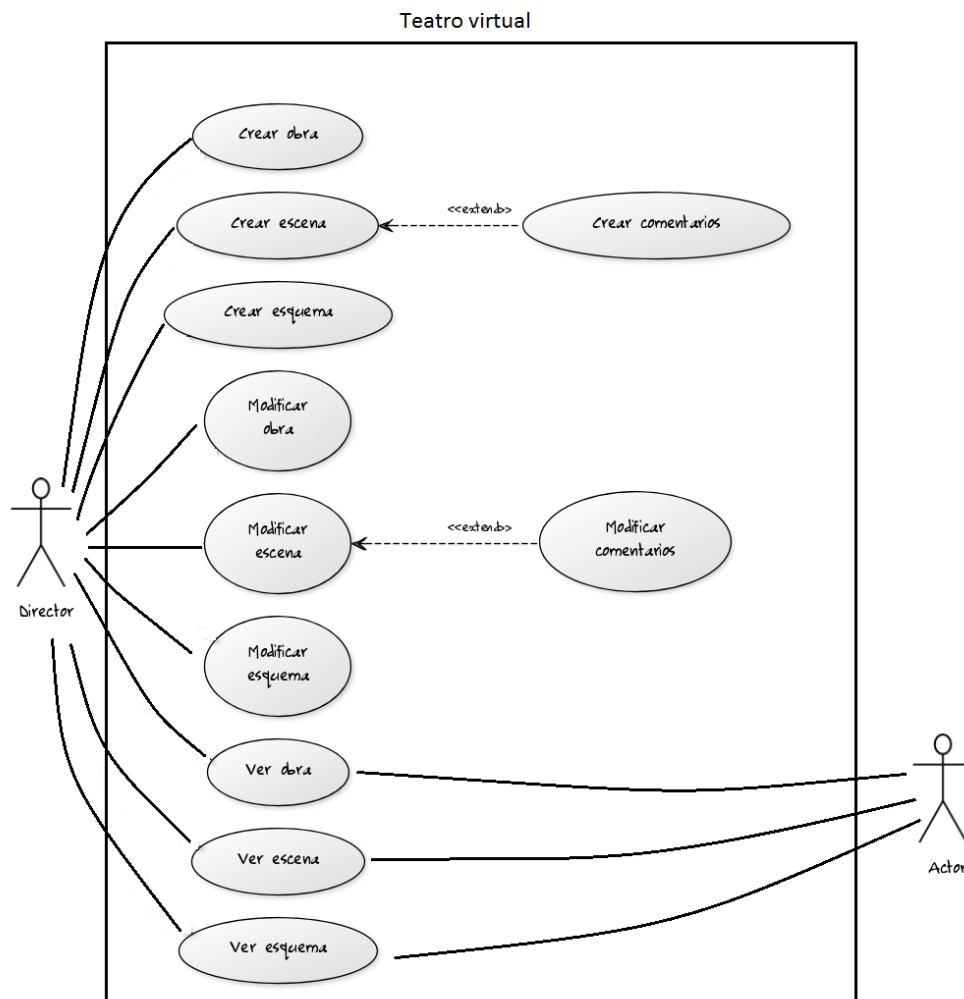
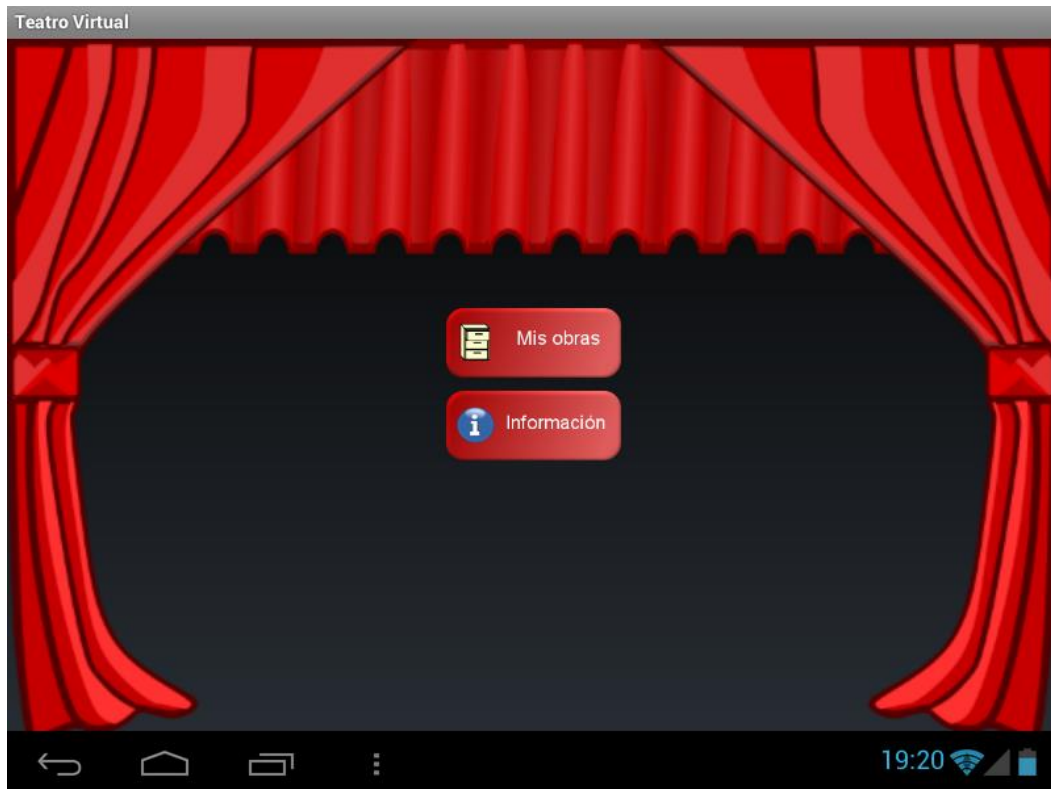


Diagrama UML de casos de uso

Inicio: Es la presentación de Teatro virtual, tiene dos botones, *Mis obras* e *Información*.



Pantalla de inicio

Información: Esta es una pantalla explicativa. En ella se encuentran tanto las instrucciones de manejo de la app como información adicional. Es importante destacar que además de estas instrucciones todas las pantallas cuentan con pequeños mensajes, (toasts) con explicaciones sobre el manejo de cada una de ellas.

Cómo usar Teatro Virtual

Pulsa *Mis obras* para acceder a la pantalla con la lista de obras creadas.

En esta pantalla puedes usar el menú de opciones para crear obras nuevas o pulsar sobre los títulos ya creados para modificarlos, ver todos los esquemas ya creados (toque largo) o acceder a las escenas de cada obra (toque corto).

La pantalla de escenas tiene un funcionamiento similar y te permite visualizar la lista de escenas de cada obra. Pulsando en cada escena accedes a un editor que te permite tomar notas.

Pulsa *Dibujar escena* para acceder a la pantalla de dibujo. En esta pantalla puedes diseñar los esquemas de movimiento de cada escena. Arrastra las figuras de la paleta hasta su posición en el escenario, guarda tus dibujos pulsando el círculo verde y ve cambiando las posiciones con la flecha de avanzar. Puedes modificar las que ya están creadas con la flecha de retroceder.

Pulsa la imagen del ojo para visualizar los esquemas y notas ya creados. Avanza entre posiciones con las flechas y entre escenas arrastrando la pantalla.

Información adicional

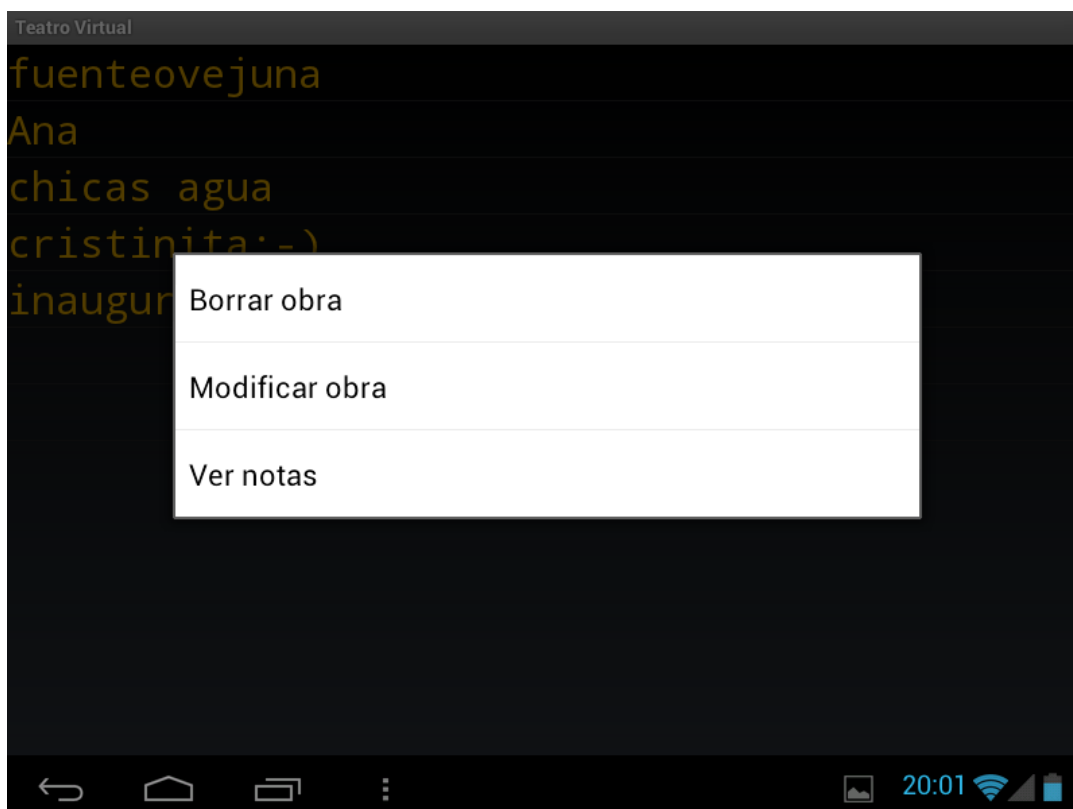
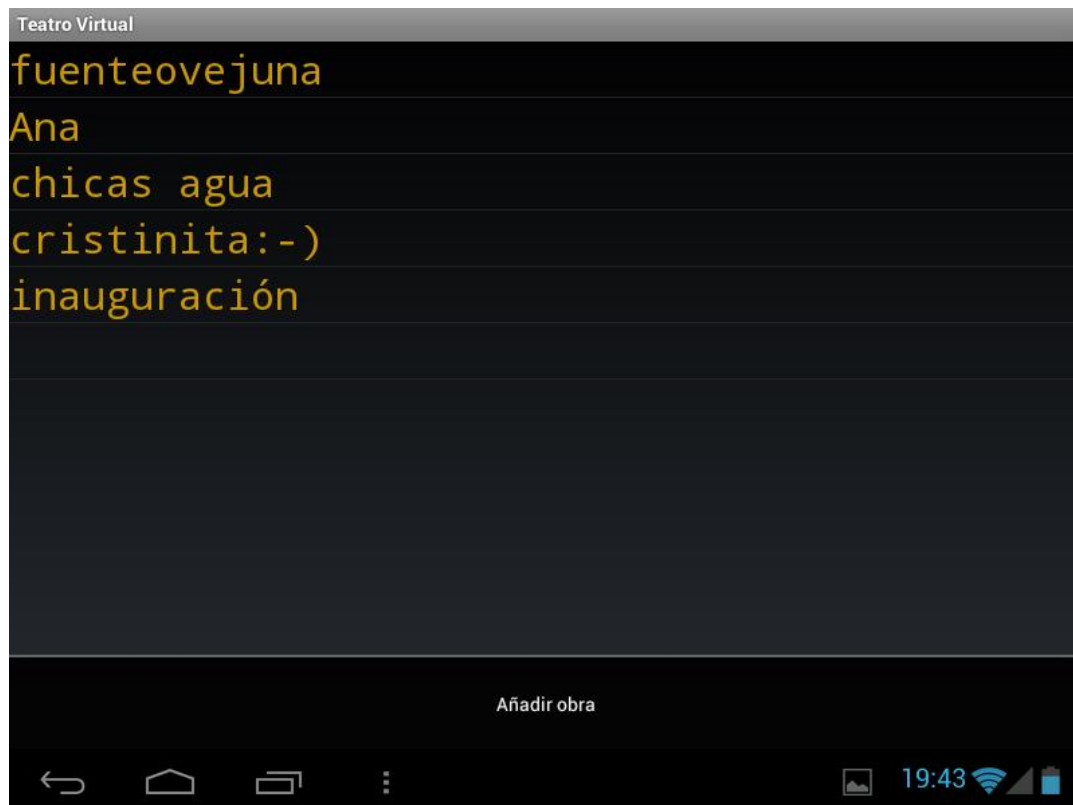
Esta aplicación ha sido desarrollada por Cristina Vidal como proyecto de fin de carrera de la UC3M. Para más información podéis contactar mandando un mail a: crisvidalnunez@gmail.com



Pantalla de información

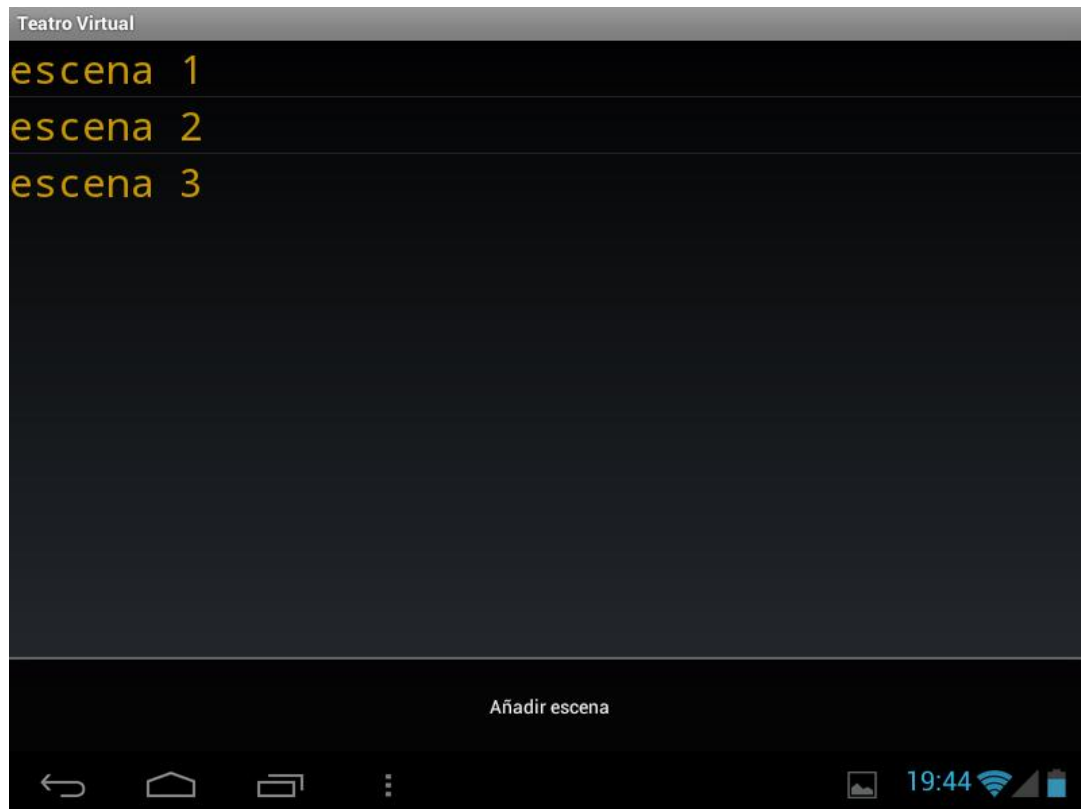
Lista de obras: Accedemos a través del botón *Mis obras*. Esta pantalla muestra una lista con las obras ya creadas. Si pulsamos sobre el título de la obra llegaremos a la pantalla donde se presentan sus escenas. Si la pulsación es larga aparecerá un menú contextual que nos permitirá eliminar la obra, (y por consiguiente sus escenas y dibujos asociados), modificarla, (accederíamos al editor de obras para cambiar su título) o ver las notas directamente (accederíamos a la pantalla final Notas)

También podemos usar el menú de opciones para crear una nueva obra; esta opción también nos lleva al editor de obras.



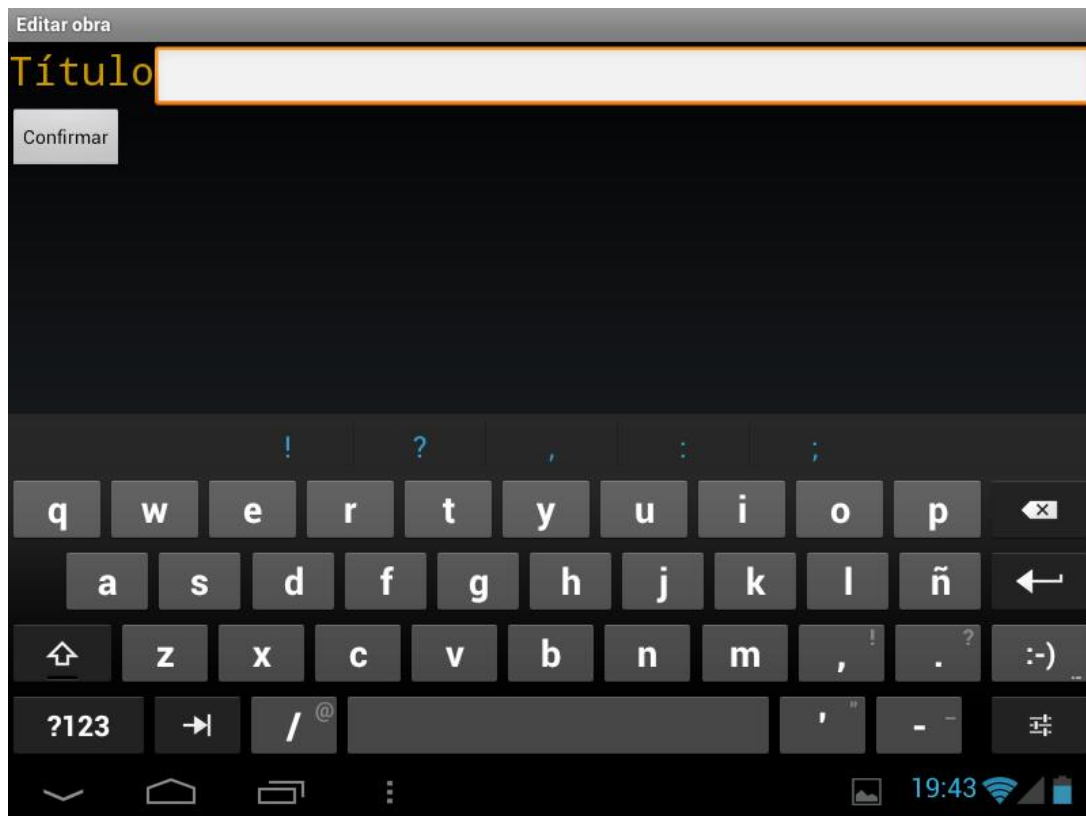
Pantalla de lista de obras

Lista de escenas: El funcionamiento de esta pantalla es muy similar al de la lista de obras. Al pulsar sobre el título de la escena llegamos directamente a la pantalla Editor de escenas. Si damos una pulsación larga aparecerá la opción Borrar escena en el menú contextual la cual eliminará no solo su título y sus comentarios sino todos los esquemas asociados. Podemos usar el menú de opciones para crear una nueva escena; esta opción también nos llevará al editor de escenas.



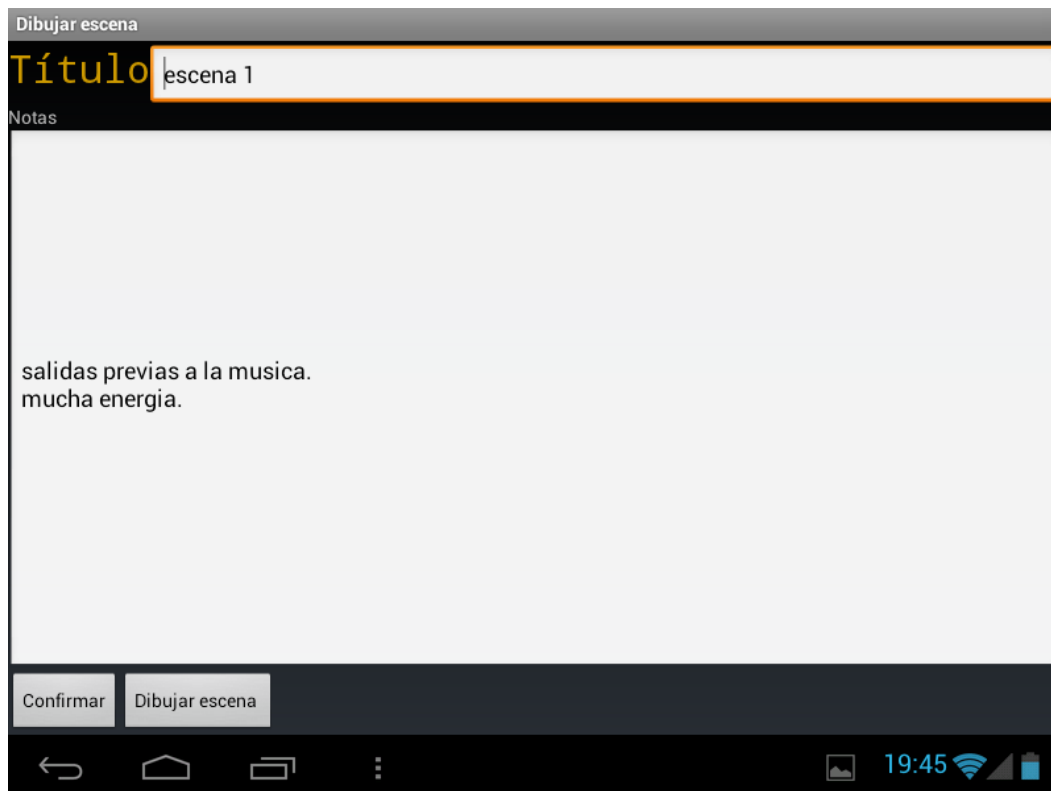
Pantalla de lista de escenas

Editor de obras: Es una pantalla muy sencilla. Cuenta con un espacio para insertar el título de la obra y un botón *Confirmar* que nos devuelve a la pantalla Lista de obras.



Pantalla del editor de obras

Editor de escenas: Cuenta con un espacio para insertar el título de la escena, otro para tomar notas y dos botones: *Confirmar*, que nos devuelve a la pantalla Lista de escenas y *Dibujar escena* con el que accedemos a la pantalla Editor de esquemas



Pantalla del editor de escenas

Editor de esquemas: Esta es la pantalla más compleja; se puede dividir en tres zonas y presenta la mayor funcionalidad de toda la app. En esta pantalla podemos crear los dibujos que ilustren cada una de las posiciones existentes en una escena, guardarlos y modificarlos.

La zona del lado izquierdo presenta una paleta con todos los iconos que pueden usarse para ilustrar una posición. La selección de los iconos fue compleja porque queríamos que éstos cumplieran todas las necesidades del director sin ser demasiado numerosos para no dificultar su tarea. Optamos por proporcionar un pequeño número de dibujos pero procurando que estos puedan representar el mayor número de conceptos posibles. Tenemos por tanto seis actores individuales, cada uno de un color diferente para que sean fácilmente reconocibles, dos coros, también de colores diferentes, un círculo que puede funcionar como cualquier elemento del decorado, cuatro rectas con distintas orientaciones que cumplen el mismo propósito (pueden usarse tanto como objetos como para acotar el espacio) y seis flechas que sirven para indicar la dirección del movimiento de los actores.

La parte superior derecha de la pantalla la ocupa el espacio del escenario, tiene unas dimensiones de 6x15 y es la zona a la que arrastraremos nuestros iconos.

En la parte inferior derecha nos encontramos con el título que previamente hemos puesto a cada escena y con varios botones. Teniendo en cuenta que a esta pantalla se accede desde cada escena pero cada escena puede contar con varias posiciones, los botones que usaremos serán los siguientes:

Papelera para descartar los iconos mal situados, botón de *guardar* para salvar las distintas posiciones, dos *flechas* para ir cambiando de posición (la aplicación solo te permite usar la flecha de avance para crear una nueva imagen una vez hayas guardado la posición actual), y el *ojo* que nos lleva a la última pantalla, Notas.

Es importante indicar que cada vez que se pulsa la flecha de avance los iconos del escenario se mantienen, ahorrando tiempo al usuario pues el esquema de final de una posición coincide necesariamente con el esquema de comienzo de la siguiente. Este es uno de los momentos donde más claramente se aprecia las ventajas de este método frente al tradicional en el que tienes que estar copiando dibujos prácticamente idénticos una y otra vez.



Pantalla del editor de esquemas

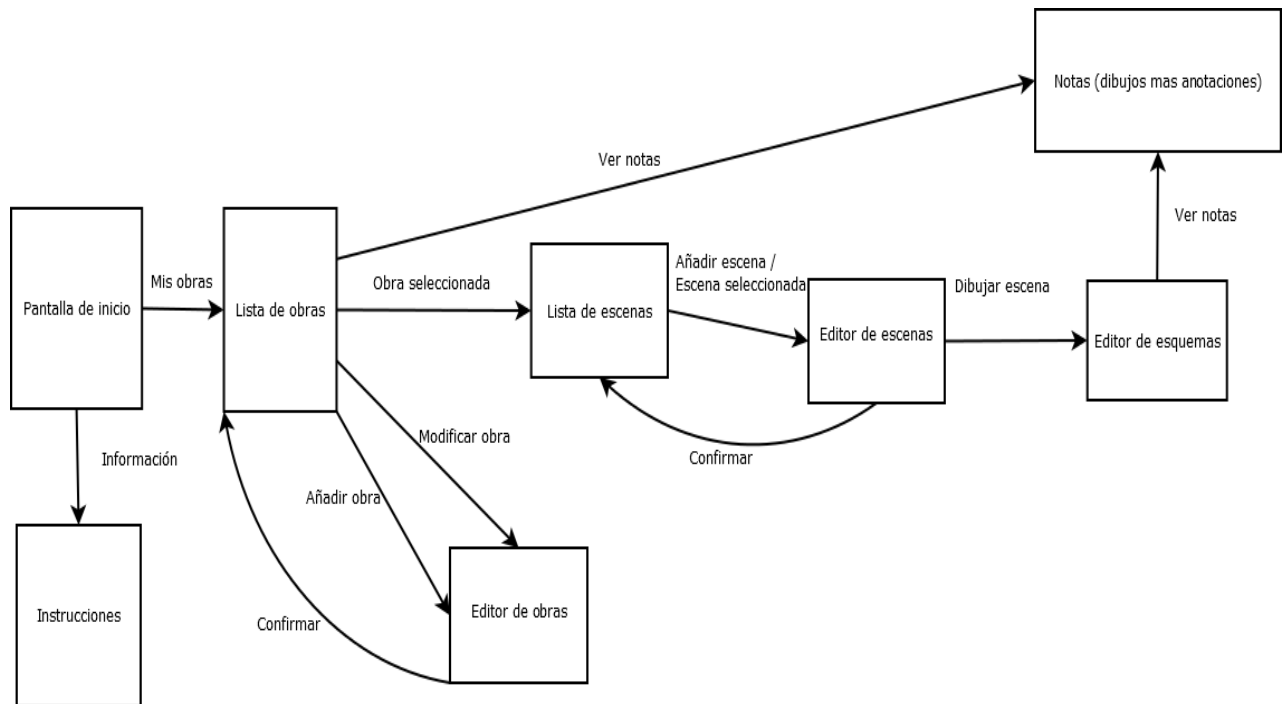
Notas: Esta es la pantalla que nos permite visualizar las obras creadas.

En la parte superior se presenta el título de la obra. Continúa por el título de la escena y le siguen los apuntes que se hayan podido realizar sobre ésta. Debajo de las anotaciones se presenta el primer dibujo creado y al final de la pantalla dos flechas que nos ayudan a transitar por las diferentes posiciones creadas. Para pasar a la siguiente escena basta con deslizar los dedos por la pantalla de derecha a izquierda.



Pantalla de notas

A continuación presentamos un esquema de las pantallas que acabamos de explicar y sus conexiones entre ellas:



Esquema de la estructura de la interfaz

Diseño arquitectónico:

La programación con Android se realiza en java ayudándose también de archivos xml y html. Todas las aplicaciones en Android presentan una misma estructura básica que se compone del código fuente en sí, archivos de recursos y vistas, librerías de código y el Android Manifest.

En este apartado explicaremos la estructura interna de nuestra aplicación comenzando por la organización de sus directorios y resumiendo brevemente las clases más importantes que luego explicaremos con más detalle en el apartado implementación.

Directorio SRC:

En este directorio se encuentra toda la lógica de la aplicación, las clases programadas en java. Dentro de él se pueden definir distintos paquetes.

En el caso de Teatro virtual decidimos agrupar las clases en 4 paquetes: **cyberteatro.activity**, **cyberteatro.model**, **cyberteatro.pantalla.drag** y **cyberteatro.utils**.

En **cyberteatro.activity** creamos las clases que se corresponden con las pantallas que maneja el usuario, (salvo la pantalla editor de esquemas que debido a su complejidad se incluye en un paquete diferente) así como alguna más de ayuda.

En **cyberteatro.model** se encuentran las clases que representan los modelos básicos que usamos en nuestra aplicación: obras (Obra.java), escenas (Escena.java), imágenes dentro de una escena (Imagen.java) y objetos que conforman una imagen (Objeto.java)

cyberteatro.pantalla.drag engloba todas las clases necesarias para crear la pantalla Editor de esquemas, que es la más compleja.

En **cyberteatro.utils** se encuentran las clases GridViewUtils.java que cuenta con los métodos estáticos responsables de cargar los objetos de una escena y la clase DbAdapter.java.

Android library:

En este paquete se encuentran todas las librerías propias del SDK de Android.

Directorio RES:

En este directorio se encuentran los archivos con los recursos que usa nuestra aplicación tales como imágenes en sus diferentes resoluciones o archivos xml donde se definen los idiomas y estilos. De este directorio es necesario destacar la carpeta Layout que contiene todos los archivos xml con las interfaces visuales asociadas a las actividades, es decir, las interfaces de las distintas pantallas de la aplicación.

Directorio bin:

Este directorio contiene los archivos generados por la propia aplicación. También se encuentra el ejecutable de la aplicación, conocido como apk.

Directorio gen:

En esta carpeta se encuentra el archivo R.class, autogenerado, que contiene los identificadores de los recursos empleados en la aplicación.

Directorio assets:

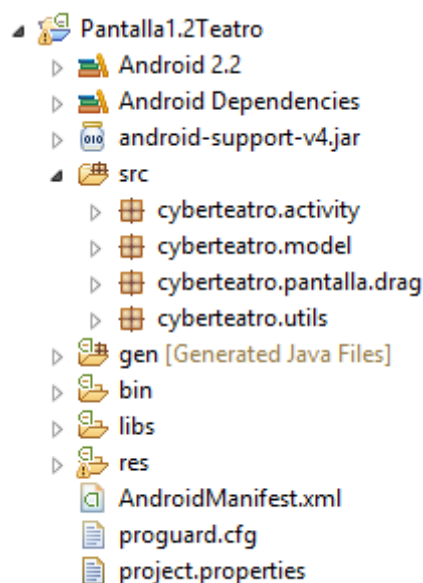
Este directorio contiene los archivos auxiliares de la aplicación a los que se accede mediante una ruta física en lugar de generar un identificador dentro del R.class. No hemos tenido necesidad de usar este directorio para nuestra aplicación.

Directorio lib:

En este directorio se instalarían las librerías internas. Tampoco se va a usar en Teatro Virtual

Android Manifest:

Es el archivo básico de configuración de cualquier aplicación Android. En él se definen todas las actividades de la app, sus permisos etc.



Estructura interna de la app Teatro Virtual

3.3 Implementación

En este apartado analizaremos la implementación de nuestra app explicando con mayor detenimiento las clases más relevantes de cada uno de los paquetes que hemos expuesto anteriormente. También hablaremos de los archivos xml más importantes y de los puntos más significativos de la dinámica de la aplicación.

cyberteatro.activity: De todas las clases que componen el paquete, cabe destacar ViewPlay.java que representa la ventana donde se consulta una obra y sus páginas asociadas. Esta clase tiene un componente ViewPager que sirve para pasar por las diferentes escenas en el tablet, como si se pasaran las páginas de un libro. El componente ViewPager también está asociado a un objeto de la clase PageChangeListener.java que detectará cuando pasemos de una escena a otra y guardará el número de escena en el que nos encontremos dentro del ViewPager.

Pantalla1_1TeatroActivity.java y Pantalla1_2TeatroActivity.java son las clases encargadas de crear las pantallas que muestran la lista de obras y lista de escenas respectivamente. Ambas se ayudan del objeto dbAdapter para acceder a la base de datos. Para este tipo de operaciones se programa un try-catch-finally donde se abre la BD en el bloque try y se cierra en el finally. Este mismo bloque se usa también en las clases PlayEdit.java y SceneEdit.java, responsables de las pantallas Editor de obras y Editor de escenas lo que les permite modificar la BD. En ambas clases destaca el método populateFields() gracias al cual podemos cargar los datos de la obra y de las escenas.

FragmentSceneAdapter.java gestiona la lista de escenas; con cada nueva escena se crea un objeto EscenaFragment con su información que se añade al objeto FragmentSceneAdapter. La clase EscenaFragment.java por lo tanto representa una escena determinada y su vista asociada es Sceneview.xml. Sus métodos más importantes son fillEscena(), run(), que se ejecuta después de cargar la pantalla, y onCreateView() que se sirve de fillEscena() para poder generar las distintas vistas de nuestras escenas.

Por último, la clase Main.java es la encargada de mostrarte la pantalla de inicio mientras que Help.java crea la pantalla de información.

cyberteatro.model: En ella se encuentran las clases que representan las obras, las escenas, las imágenes pertenecientes a una escena y los objetos que usan dichas imágenes.

cyberteatro.pantalla.drag: engloba todas las clases necesarias para crear la pantalla Editor de esquemas, que es la más compleja.

Entre ellas destacamos la clase `DragActivity.java` que define el área, en forma de grid, de la pantalla donde se pueden posar los iconos, así como toda la lógica necesaria para guardar y cargar los dibujos de las distintas posiciones.

`DeleteZone.java`, es la encargada de describir el área, dentro del `DragLayer` donde se puede posar un objeto arrastrado para eliminarlo de la pantalla. Al ser una subclase de `ImageView` podremos representarla en la pantalla como una imagen, en este caso de papelería de reciclaje.

La clase `DragController.java` se encarga de manejar eventos táctiles arrastrando una view dentro de un `viewgroup`.

`DragView.java` es una vista especial que se usa durante las operaciones de drag. La imagen que el usuario ve moverse por la pantalla en esas ocasiones es en realidad un bitmap creado por esta clase.

`DragSource.java`, por su parte, representa el interfaz encargado de definir un objeto donde comienzan las operaciones de drag.

`ImageCell.java` es una subclase de `ImageView` usada para mostrar una imagen en nuestro `GridView`. Las `Image Cells` representan los lugares hacia donde se pueden arrastrar los iconos o desde donde se pueden mover por lo que esta clase implementa las interfaces `DragSource` y `DropTarget`.

Por su parte, la clase `ImageCellAdapter.java` es el adaptador para las celdas del grid donde creamos las imágenes. Dentro de esta clase cabe destacar el método `getView()` que devuelve un objeto tipo `View` para poder usarlo en el grid.

En la clase `DropTarjet.java` programamos la interfaz que define un objeto que reacciona ante objetos (`dragViews`) que son arrastrados por encima o echados sobre él. Esta clase se sirve además de los métodos `acceptDrop()` al que se llama repetidamente durante el arrastre para saber si en la zona por la que pasa el objeto se puede realizar un drop y el método `estimateDropLocation` que estima el área en la que aterrizaría el objeto si se soltase en ese instante().

cyberteatro.utils: se encuentran las clases `GridViewUtils.java` que cuenta con los métodos estáticos responsables de cargar los objetos de una escena y la clase `DbAdapter.java`.

La clase `GridViewUtils.java` cuenta con el método `LoadImageToScreen()`, que carga los objetos en una nueva escena y donde se programa la lógica responsable de mantenerlos ahí para crear una nueva posición dentro de la misma escena. Gracias al método `cleanScreen()` el grid se limpia antes de cargar una nueva imagen y con el método `SaveImageFromScreen()` guardamos todos los objetos de la escena en la base de datos.

Dentro de DbAdapter.java se encuentra toda la lógica de la base de datos. Esta clase define las operaciones básicas CRUD (create, read, update and delete). Para programar DbAdapter nos servimos de SQLite. Fuimos creando diferentes tablas: de obra, de escenas y de objetos que se encuentran en cada escena.

Los métodos más destacados de esta clase son onCreate(), que se invoca la primera vez que se crea la base de datos, fetchPaintFromScene() encargado de devolver un cursor con todos los objetos de una imagen de una escena, getCountPaintFromScene() que devuelve el número de dibujos de una escena y updateObject() para modificar un objeto.

Es importante también hablar de los archivos xml más destacados que encontramos en la carpeta de recursos. Estos archivos se utilizan para declarar layouts y demás elementos de los que nuestra aplicación necesita hacer uso.

Cada archivo xml está formado por un árbol de elementos que especifica cómo los elementos de la UI y los contenedores se acomodan para definir la parte visual de un objeto View.

Dentro de la carpeta Layout encontramos los archivos XML encargados de dar forma a cada una de nuestras pantallas.

En la carpeta values (perteneciente también a Res) nos encontramos con otros tres archivos que usamos para apuntar el diseño de nuestra aplicación de forma más general; En estilos.xml definimos el tipo, color y tamaño de los textos que aparecen en las pantallas principales. En mycolor.xml asociamos las etiquetas de colores que aparecen en todas las clases java, con sus colores correspondientes en base hexadecimal, (colores de las celdas vacías o llenas de nuestro grid, de las imágenes que arrastramos, de los fondos etc.). Para finalizar, en strings.xml nos encontramos con las etiquetas para definir todos los textos que aparecen en nuestra aplicación, toasts incluidos. Cabe destacar el texto de la pantalla de ayuda que al ser el más largo definiremos empleando también etiquetas html.

Decidimos definir estas estructuras vía XML en lugar de implementarlo directamente desde el código java porque es más práctico tener separado código de diseño y código de funcionalidad a la hora de modificar la interfaz de usuario. También es ventajoso tener separado el código por defecto de la aplicación del que se genera en respuesta a alguna función realizada por el usuario.

3.4 Pruebas

Una vez programada la estructura de la aplicación, con toda su lógica y una versión de la interfaz lo más similar posible a la final, llega el momento de probar la app para testar su validez e ir corrigiendo los pequeños fallos que vayamos encontrando a nuestro paso.

La fase de pruebas consta de dos etapas; la primera la realiza el propio programador probando cada una de las opciones disponibles en la aplicación y la segunda se lleva a cabo a modo de encuesta, permitiendo a usuarios que no han estado implicados en el proceso de creación probar la app y compartir sus opiniones.

Pruebas internas:

Consisten en recorrer la aplicación probando todo aquello para lo que se ha diseñado, para empezar creamos una obra con varias escenas, dentro de cada escena dibujamos varias posiciones y una vez guardada tratamos de modificar cada una de sus partes y también de eliminarlas.

Descubrimos pequeños fallos como que no se puede modificar el título de la obra o que se produce un error si en la pantalla Editor de escenas pulsas el botón dibujar antes que el de guardar. Una vez solventados estos errores pasamos a la siguiente fase, las pruebas externas.

Pruebas externas:

Como explicamos anteriormente, hemos realizado una encuesta a un pequeño grupo para que nos transmitan sus impresiones acerca de la aplicación. Este grupo de estudio lo formaron 15 personas; cabe destacar que 10 de ellas son actores y/o directores en activo y 3 son ingenieros por lo que consideramos que se completan satisfactoriamente los dos acercamientos existentes en este proyecto. Es importante también indicar que dentro del grupo de actores, un porcentaje muy elevado no había manejado nunca una tablet.

Tras pasar una media de 10 minutos manejando la aplicación nos dieron su opinión basándose (aunque sin estar obligados a contestar a cada una por separado) en las siguientes preguntas orientativas que exponemos a continuación:

- ¿Consideras a Teatro Virtual una aplicación intuitiva?

El 100% de los encuestados consideró a Teatro Virtual una herramienta intuitiva y de fácil manejo. Los únicos problemas que encontraron algunos de los usuarios fueron relativos al uso de la propia

tablet al no estar familiarizados con su interfaz; esas dificultades se plasmaron a la hora de acceder a los menús conceptuales y al usar la flecha que te lleva a la anterior pantalla. Aun así es importante destacar que se ha cumplido el objetivo que mencionábamos en el capítulo uno de que esta aplicación debía orientarse a personas que no tenían por qué tener conocimientos informáticos avanzados ya que la curva de aprendizaje de los encuestados que no habían usado nunca una tablet fue muy rápida, tanto a nivel general con la interfaz del dispositivo como a nivel particular con la aplicación Teatro Virtual.

Consideramos fundamental que la respuesta a esta pregunta sea afirmativa, pues a pesar de que existen unas instrucciones, accesibles desde la pantalla inicial, es importante que el programa se entienda por sí mismo ya que la tendencia general de los usuarios es la de empezar a manejar la aplicación inmediatamente, sin tener que verse obligados a pasar por esa primera pantalla explicativa.

➤ ¿Te parece una aplicación práctica?

Esta pregunta también se contestó afirmativamente y con unanimidad. Es importante recordar que aproximadamente la mitad de las personas que realizaron esta encuesta habían participado en la reunión inicial en la que terminaron de definirse los objetivos de la aplicación y las necesidades del sector. El resto del grupo de prueba que no participó en la reunión y que siendo actores o directores emplean una metodología diferente a la hora de crear obras, menos clásica y más orientada a la creación colectiva, también encontraron útil la aplicación por lo que damos este objetivo por cumplido.

➤ ¿La interfaz te resulta agradable?

La interfaz resulta agradable aunque hubo un porcentaje de usuarios que pensaron que podría mejorarse. De hecho, la versión que presentamos en esta memoria es una versión posterior a la encuesta y cuenta con ligeras modificaciones para que se acerque más al gusto de los usuarios. Básicamente reformamos la pantalla Editor de esquemas: cambiamos los iconos de los actores y del coro para volverlos más sofisticados y modificamos el botón que nos envía a la pantalla Notas añadiéndole texto porque para algunas personas resultaba confuso. También cambiamos el font de las pantallas de obras y escenas para mejorar la línea de estilo.

➤ ¿Qué elementos mejorarías o añadirías?

Como hemos comentado en la pregunta anterior una parte de los encuestados mejorarían la interfaz. Tras cambiar varios iconos de la pantalla Edición de esquemas añadimos otros nuevos a petición suya como son las flechas. Estos 6 iconos ayudarán a los directores a indicar la dirección y sentido del movimiento de los actores.

Un pequeño porcentaje de los usuarios sugirió añadir la opción de poner música a las escenas, de permitir que los iconos de los actores pudieran modificarse para ponerles etiquetas con nombres, crear una biblioteca más completa con todo tipo de objetos o contar con la posibilidad de poder exportar los esquemas creados en formato .jpeg o similares.

Vemos como algunas de estas sugerencias no casan con la idea de crear una aplicación sencilla y manejable; otras son interesantes pero caen fuera del alcance de este proyecto por lo que las guardaremos para estudiarlas en el apartado líneas de trabajo futuras.

➤ ¿Qué elementos suprimirías?

Los encuestados no suprimirían ninguno de los elementos que se encuentran actualmente en la aplicación.

Capítulo 4

Conclusiones y trabajos futuros:

4.1 Conclusiones

Tal y como hemos expuesto a lo largo de este trabajo, las apps dedicadas a los montajes teatrales suponen un gran avance para el sector gracias a que el usuario puede disponer de un mayor número de herramientas de forma más rápida y sencilla que con los métodos tradicionales.

Si bien todavía queda mucho camino por recorrer, en este proyecto se han cumplido los objetivos expuestos en el capítulo 1; hemos hecho un estudio acerca del estado del arte de los dispositivos Android, sus aplicaciones y la relación entre la industria teatral y las nuevas tecnologías. Una vez establecido en qué punto se encuentra esa relación y estudiado diferentes formas en las que el sector podría beneficiarse, diseñamos e implementamos nuestro Teatro Virtual.

Al usar dispositivos Android, ofrecemos a sus usuarios una herramienta más asequible que las proporcionadas por el iOS de Apple, y que todavía no cuenta con tantas aplicaciones sobre el tema.

En resumen, la creación de esta app nos ha permitido realizar una pequeña contribución hacia la apertura de un camino digital para el sector teatral, más eficaz, interactivo y también más económico.

4.2 Trabajos futuros

Las líneas de trabajo futuras que surgen tras la implementación de Teatro Virtual se centran básicamente en el perfeccionamiento de la aplicación y en el incremento de sus prestaciones. Estas mejoras se pensaron tanto durante la fase de estudio del estado del arte como durante la fase de pruebas, tras escuchar las opiniones y sugerencias de sus usuarios potenciales.

A continuación expondremos las mejoras más significativas y las futuras líneas de desarrollo.

➤ Mejora de la interfaz:

Si bien la aplicación creada es muy funcional, sería interesante dedicarle más tiempo a la línea de estilo que presentan las pantallas mejorando el diseño gráfico de las mismas. Detalles como hacer la pantalla de instrucciones o los iconos más sofisticados u optimizar las dimensiones de las diferentes zonas en la pantalla Editor de esquemas ayudarían a la hora de lanzar este producto al mercado.

➤ Nuevas funcionalidades:

A continuación expondremos varias ideas que fueron surgiendo a lo largo de las fases de diseño, implementación y pruebas las cuales permitirían añadir nuevas funcionalidades al programa y mejorar las ya existentes en una segunda fase de desarrollo siempre y cuando no hicieran excesivamente complejo el manejo de la interfaz:

- Posibilidad de personalizar los iconos de los actores, coros y escenografía para, por ejemplo, añadir una etiqueta de texto, modificar su tamaño etc, (la opción más básica debería estar también disponible para, como hemos explicado anteriormente, no añadir una complejidad innecesaria)
- Añadir una opción *pintura* en la paleta de iconos de la pantalla Editor de esquemas. Haciendo que la planta del teatro también funcione a modo de canvas, el usuario podría añadir sus propios dibujos e indicaciones sobre los esquemas.
- Posibilidad de seleccionar diferentes tipos de plantas del escenario. Junto con el canvas, ésta es una de las escasas funcionalidades que presenta la app mencionada en el capítulo 2, Theater Blocking. Si bien nuestro Teatro Virtual nos permite acotar, en cierta forma, el espacio usando los iconos de pared, sería interesante contar con fondos que presenten distintas vistas cenitales donde se pudiera incluir el espacio del público o plantas de escenarios poco convencionales.
- Guardar los esquemas en formato .JPEG o similares y crear un directorio dentro de la aplicación que permita enviar al correo de los actores y demás miembros del equipo esas imágenes para que éstos puedan visualizarlas incluso sin tener instalada la aplicación.

Capítulo 5

Presupuesto:

En este capítulo calculamos el coste de desarrollo de la aplicación diferenciando entre costes de personal, de material e indirectos.

5.1 Costes de personal

Para calcular los costes de personal es necesario realizar una estimación del tiempo empleado en cada una de las fases del proyecto, tarea difícil pues debido a la necesidad de compatibilizar trabajo con estudios y diferentes proyectos la realización de este programa se ha dilatado bastante en el tiempo. Haremos pues, una estimación del tiempo neto empleado en cada fase suponiendo que cada mes cuenta con 21 días laborables y jornadas de 8 horas diarias. Para realizar el cálculo se ha considerado el salario medio de un ingeniero según las plantillas de la universidad, esto es, 2694´39 euros por persona al mes.

Fases	Dedicación en meses	Coste en euros
Estudio de las tecnologías	2	5388,78
Diseño de la aplicación	1	2694´39
Implementación de la aplicación y pruebas	3	8083,17
Escritura y corrección de la memoria	1,5	4041,58
Total	7,5	20207,9

5.2 Costes de material

El material que hemos usado para el desarrollo de este proyecto se resume en un ordenador portátil personal valorado en 600 euros, (considerando que estos

aparatos tienen una vida media de cuatro años y que hemos trabajado con él durante siete meses sumaremos al proyecto 93,75 euros por su uso) para la programación y una tablet Android valorada en 149 euros para poder estudiar el funcionamiento y las interfaces de las aplicaciones Android y para poder probar la nuestra en un entorno real. Dado que todo el software necesario para la elaboración del programa es libre, en los costes de material solo se incluirá el hardware, ascendiendo este a 242,75 euros.

5.3 Costes indirectos

Los costes indirectos se calculan como el 20% del coste total y engloban todos aquellos gastos como electricidad, conexión a internet, etc. Estos costes ascienden a 4041,58 euros.

5.4 Total

Concluimos pues calculando el presupuesto total que en nuestro caso es de 24492,23 euros.

Capítulo6

Anexos:

6.1 Manual de uso

6.1.1 Instalación

Abre el fichero .apk proporcionado por el programador en cualquier dispositivo Android y éste se instalará automáticamente.

6.1.2 Instrucciones de Teatro Virtual

Pulsa *Mis obras* para acceder a la pantalla Lista de obras. En esta pantalla puedes usar el menú de opciones para crear obras nuevas o pulsar sobre los títulos ya creados para modificarlas, ver los esquemas ya creados (toque largo) o acceder a las escenas de cada obra (toque corto).

La pantalla Lista de escenas tiene un funcionamiento similar y te permite visualizar las escenas de cada obra. Pulsando en cada escena accedes a un editor que te permite tomar notas. Pulsa *Dibujar escena* para acceder a la pantalla de dibujo.

En la pantalla Editor de esquemas puedes diseñar los esquemas de movimiento de cada escena. Arrastra las figuras de la paleta hasta su posición en el escenario, guarda tus dibujos pulsando el círculo verde y ve cambiando las posiciones con la flecha de avanzar. Puedes modificar las que ya están creadas con la flecha de retroceder. Pulsa la imagen del ojo para visualizar los esquemas y notas ya creados. Avanza entre posiciones con las flechas y entre escenas arrastrando la pantalla.

6.2 Diagrama de clases

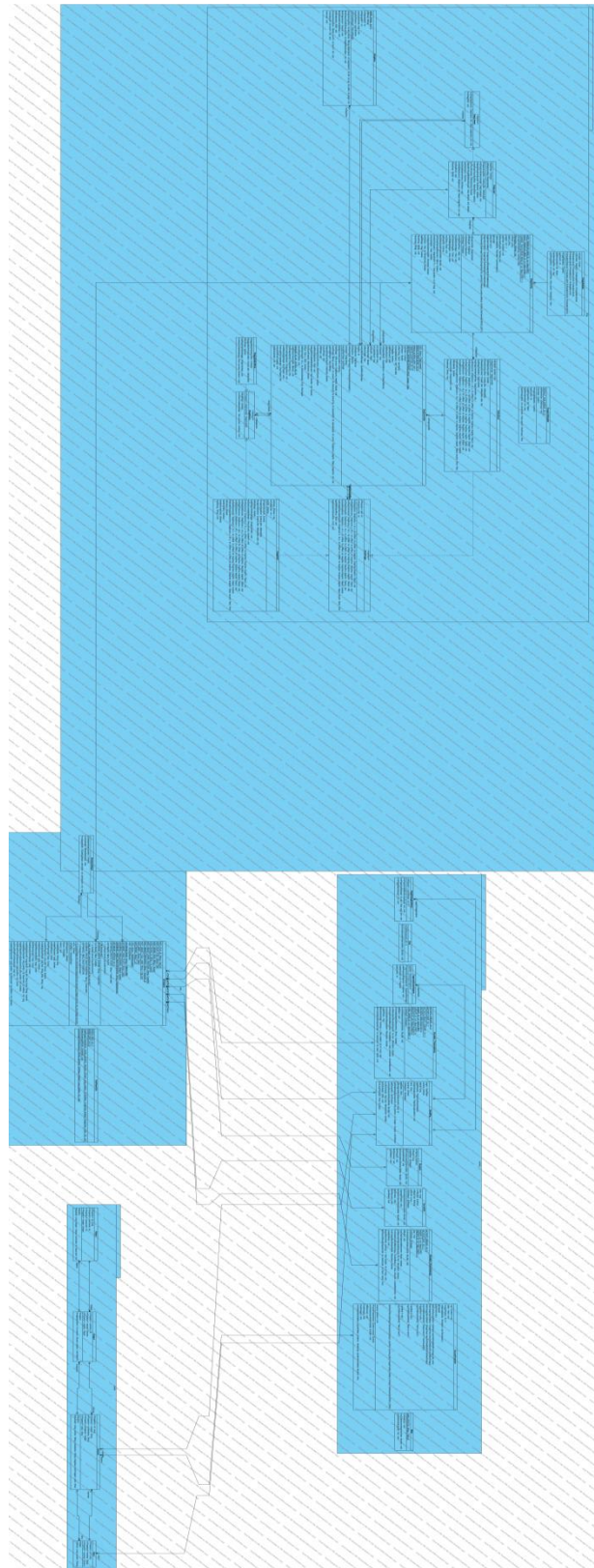
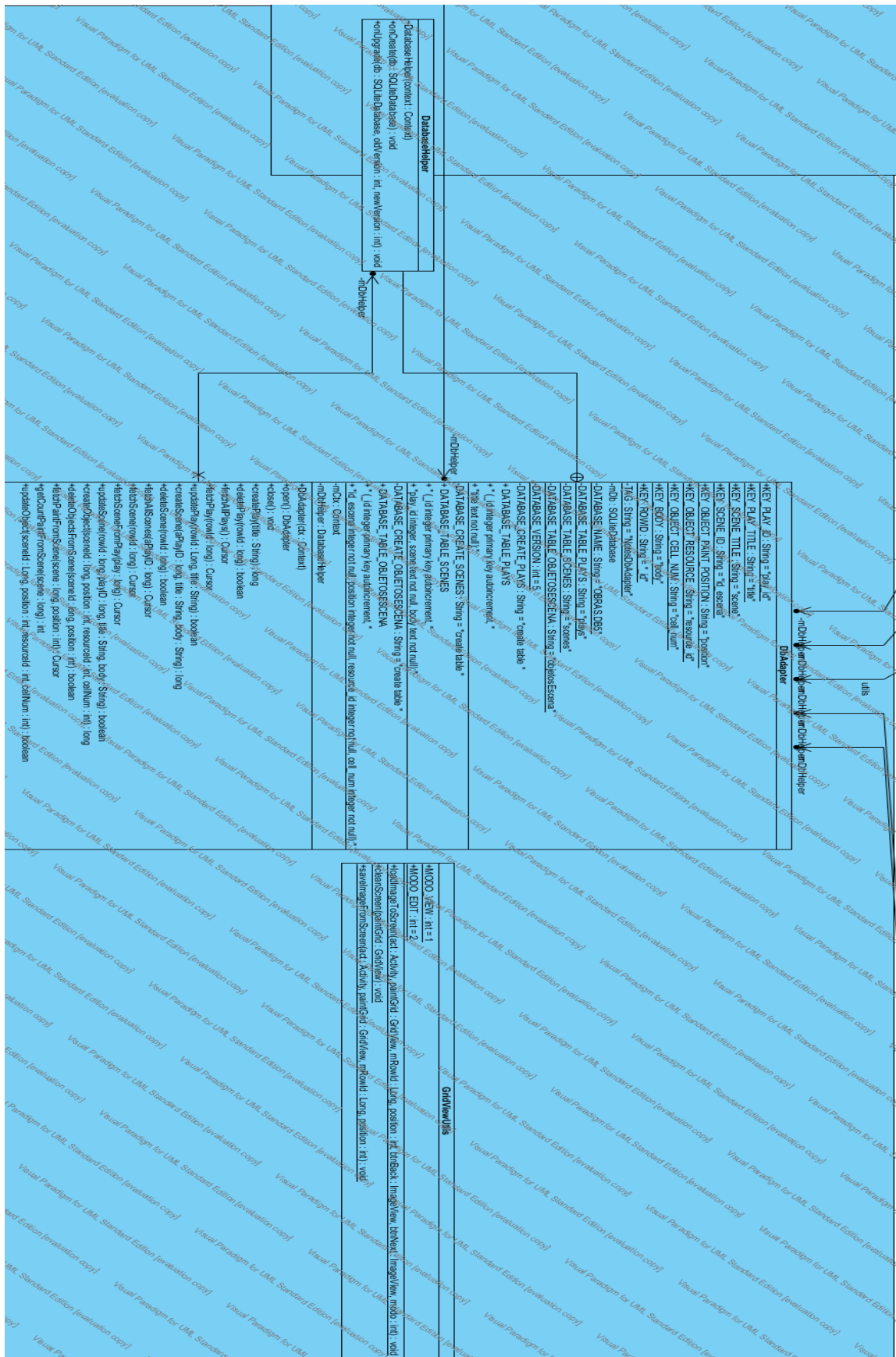
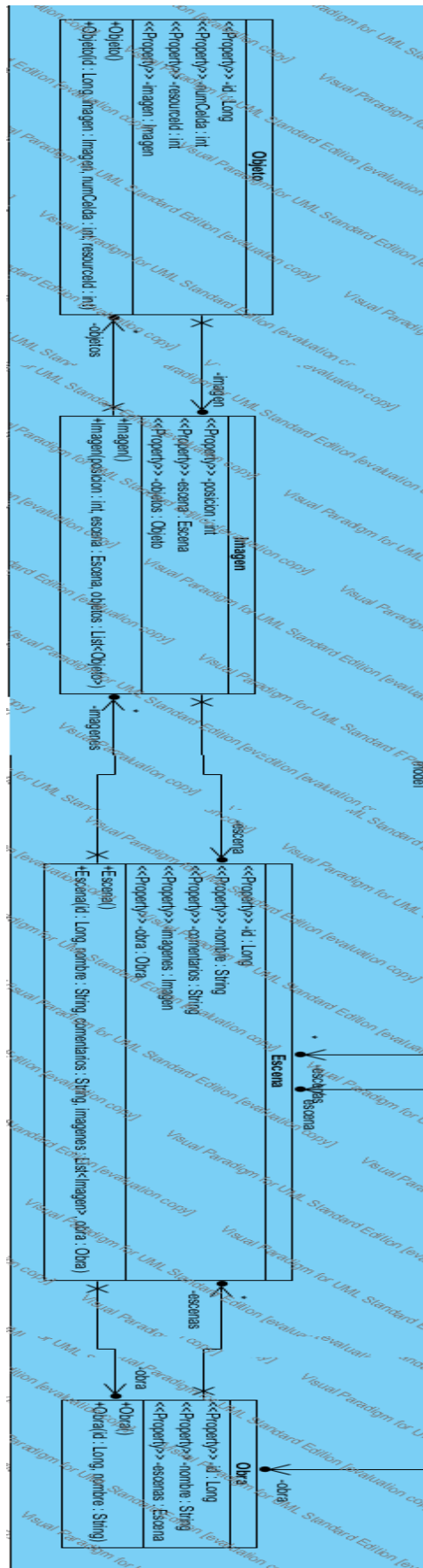


Diagrama de clases dividido de izda. a dcha. de arriba abajo en paquete 1(1a y 1b), paquete 2, paquete 3, y paquete 4.



Paquete 2 cyberteatro.utils



Paquete 4 cyberteatro.model

6.3 Glosario

Cuarta pared: Pared invisible e imaginaria que se encuentra en un escenario, a través de la cual el público contempla el espectáculo.

Apple: Empresa multinacional estadounidense que diseña y produce equipos electrónicos y software. Entre los productos de hardware más conocidos de la empresa se cuenta con equipos Macintosh, el iPod, el iPhone y el iPad. Entre el software de Apple se encuentran el sistema operativo Mac OS X, el sistema operativo iOS, el explorador de contenido multimedia iTunes

Android: Sistema operativo basado en Linux, diseñado principalmente para dispositivos móviles con pantalla táctil como teléfonos inteligentes o tablets inicialmente desarrollados por Android, Inc y que ahora pertenece a Google.

iOS: Sistema operativo móvil de la empresa Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS), siendo después usado en dispositivos como el iPod Touch, iPad y el Apple TV

Google Play: Tienda de software en línea desarrollada por Google para los dispositivos con sistema operativo Android.

HTML: HyperText Markup Language. Lenguaje de marcado hipertextual estándar que define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web.

XML: EXtensible Markup Language. Lenguaje de Marcas Extensible, es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.

Android Manifest: Archivo donde se define la configuración básica de las aplicaciones Android.

SQLite: Sistema de gestión de bases de datos relacional compatible con ACID, contenida en una pequeña biblioteca escrita en C. SQLite es un proyecto de dominio público.

ACID: Atomicity, consistency, Isolation and Durability. Atomicidad, Consistencia, Aislamiento y Durabilidad. Características que deben tener las bases de datos para que sus instrucciones sean consideradas como una transacción.

SDK: Software Development Kit. Kit de Desarrollo de Software. Conjunto de herramientas de desarrollo de software que permite al programador crear aplicaciones para un sistema concreto.

APK: Application Package File. APK es un formato de archivo que se usa para empaquetar aplicaciones para el sistema operativo para móviles Android.

Activity: Componente de una aplicación Android encargada de mostrar una pantalla en la app con la que los usuarios puedan interactuar. A cada actividad se le asigna una ventana en la que crear su interfaz de usuario.

Ice Cream Sandwich: Versión 4.0 de Android.

Canvas: Elemento que presenta una región donde el usuario puede dibujar libremente.

Blocking: Nombre con el que se conoce el proceso de planificar, dónde, cuándo y cómo se mueven los actores por el espacio durante una obra. El término fue acuñado por W.S Gilbert, quien usaba pequeños bloques de madera con tal propósito.

Toast: Componente de Android que se usa para mostrar pequeñas notificaciones en pantalla, (avisos o información para el usuario de las activities) que desaparecen en poco tiempo.

UML: Unified Modeling Language. Lenguaje Unificado de Modelado, es un lenguaje de modelado que permite la representación conceptual y física de un sistema.

Bibliografía:

PFCs:

- [1] Teatro y nuevas tecnologías: Conceptos básicos. Anxo Abuín González (2008).
- [2] Teatro contemporáneo y medios audiovisuales. Itziar Zorita Aguirre (2010).
- [3] Hacia un estado de la cuestión sobre el teatro y nuevas tecnologías en España. José Romera Castillo (2008)

ENLACES WEB:

- [4] Ciber-Teatro. <http://dimeo.wordpress.com/ciber-teatro/> (último acceso: Junio 2011)
- [5] CiberDrama. <http://es.wikipedia.org/wiki/Ciberdrama> (último acceso: Junio 2011)
- [6] Teatro Virtual. http://www.dogonefff.org/main_teatrevirtual_es.html (último acceso: noviembre 2012)
- [7] Teatro, unidades y objetivos. <http://eva-teatro.blogspot.com.es/2010/06/unidades-y-objetivos-primera-parte.html> (último acceso: noviembre 2012)
- [8] The virtual theatre Project, Ohio State University. http://accad.osu.edu/research/virtual_environment_htmls/virtual_theatre.htm (último acceso: junio 2011)
- [9] Blocking. http://en.wikipedia.org/wiki/Blocking_%28stage%29 (último acceso: noviembre 2012)
- [10] Blocking. <http://home.ipa.net/~mc64802/amtblock.html> (último acceso: noviembre 2012)
- [11] Historia del teatro. <http://en.wikipedia.org/wiki/Theatre> (último acceso: junio 2011)
- [12] Rehearsal. <http://www.rehearsaltheapp.com/> (último acceso: septiembre 2013)

- [13] ScriptWrite. <https://itunes.apple.com/us/app/scriptwrite/id334928503?mt=8> (último acceso: septiembre 2013)
- [14] Sides. <https://itunes.apple.com/us/app/sides-free/id379120749?mt=8> (último acceso: septiembre 2013)
- [15] ScenePartner. <https://www.scenepartnerapp.com/> (último acceso: septiembre 2013)
- [16] StageWrite. <http://www.stagewritesoftware.com/> (último acceso: septiembre 2013)
- [17] Android. <http://www.android.com/> (último acceso: septiembre 2013)
- [18] Android, Wikipedia. <http://es.wikipedia.org/wiki/Android> (último acceso: septiembre 2013)
- [19] Android. <http://www.configurarequipo.com/doc1107.html> (último acceso: septiembre 2013)
- [20] Arquitectura de Android. <http://androideity.com/2011/07/04/arquitectura-de-android/> (último acceso: septiembre 2013)
- [21] Arquitectura de Android. <http://columna80.wordpress.com/2011/02/17/arquitectura-de-android/> (último acceso: septiembre 2013)
- [22] Eclipse. <http://www.eclipse.org/> (último acceso: enero 2013)
- [23] Eclipse, Wikipedia. http://es.wikipedia.org/wiki/Eclipse_%28software%29 (último acceso: enero 2013)
- [24] Android y SQLite. <http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=537> (último acceso: noviembre 2012)
- [25] SQLite, Wikipedia. <http://es.wikipedia.org/wiki/SQLite> (último acceso: septiembre 2013)
- [26] Android & SQLite database. www.vogella.com (último acceso: enero 2013)
- [27] SQLite. <http://www.sqlite.org/> (último acceso: septiembre 2013)
- [28] Android SQLite database and content provider tutorial. <http://www.vogella.com/articles/AndroidSQLite/article.html> (último acceso: enero 2012)

- [29] Aplicación android con acceso a base de datos SQLite.
<http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=537>
(último acceso: septiembre 2013)
- [30] Top 5 best apps for drama and Theater.
<http://www.appconsumer.com/theatre-iphone-apps-the-art-of-sound/> (último acceso: septiembre 2013)
- [31] Open clip art. www.openclipart.org (último acceso: enero 2013)
- [32] Apps estadísticas. <http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats/e#lotsofapps> (último acceso: diciembre 2013)
- [33] Visual Paradigm. <http://www.visual-paradigm.com/> (último acceso: enero 2014)

